IJACSA
COMPUTER SCIENCE JOURNAL
International Journal of Advanced Computer Science and Applications

Volume 17 Issue 9
September 2026



ISSN 2156 - 5570 (Online)
ISSN 2158 - 107X (Print)

THE SCIENCE AND INFORMATION ORGANIZATION

www.ijacsa.thesai.org

Table of Contents

Paper 1: Adversarial Asymmetric Agents for Network Intrusion Detection Using Deep Reinforcement Learning

Authors: Curtis Rookard

PAGES 1 - 11

Paper 2: Assisted Robot Arm with Voice Recognition for Digital Healthcare

Authors: Ting Zhang, Oanh Phan, Jiang Lu

PAGES 12 - 23

Paper 3: When Agents Delegate: Accountability Chains in Multi-Agent AI Systems

Authors: Ornella Bahidika

PAGES 24 - 31

Paper 4: An Experimental Study on Credit Score Prediction Using K-Nearest Neighbors

Authors: Taylor Stonelake

PAGES 32 - 36

Paper 5: Leveraging Ancestral Health Records for Early Prediction of Cardiovascular Risk in Future Generations

Authors: Suresh Kurumalla, Bal Virdee, Ashish Khanna, Siva Shankar S

PAGES 37 - 49

Paper 6: Design and Verification of an Interpretable Low-Code Maintenance Decision Support Application for Automotive Assembly

Authors: Adriana ALDEA, Adriana FLORESCU, Catrina CHIVU

PAGES 50 - 59

Paper 7: OptiGeoRisk: A Spatially Localized and Optimized Geo-XGBoost Framework for Credit Default Risk Classification

Authors: Claher Prastian, Alexander Agung Santoso Gunawan

PAGES 60 - 71

Paper 8: Collaborative Development Method for Improving Uncertainty Management in Agile Software Development

Authors: Maha Makkass, Youness Laghouaouta, Adil Anwar

PAGES 72 - 84

Paper 9: Smart IoT Robot Toy with AI-Based Safety Judgment and Parental Monitoring Features

Authors: Check-Yee Law, Wee-Yang Goh, Yong-Wee Sek

PAGES 85 - 91

Paper 10: Real-Time Anomaly Detection Using ML, DL, and XAI for Financial Fraud: A Systematic Review

Authors: Padilla-Lopez Jimmy Andres, Lazo-Ordoñez Pedro Antonio, Juan Jesús Soria-Quijaite

PAGES 92 - 103

Paper 11: Fusion of Large Language Model and Knowledge Graph for Intelligent Power-System Q&A

Authors: Zijie Meng, Xinlei Cai, Lizhou Jiang, Kai Dong, Bingzhao Zhu, Junhong Guo

PAGES 104 - 116

Paper 12: Adapting Ephemeral ECC Authentication for Decentralized Web 3.0: A Threat Model and Security Analysis

Authors: Adarsh S V Nair, Rathnakar Achary

PAGES 117 - 133

Paper 13: Automated Generation of Arabic Procedural Court Orders in the Moroccan Legal Context

Authors: Saliha Yassine, Mohammed Essoujaa, Mustapha Esghir, Ouafaa Ibrihich

PAGES 134 - 141

Paper 14: Semi-Supervised Rice Leaf Disease and Pest Classification Using IoT Field Images

Authors: Abdul Azis, Abdul Fadlil, Tole Sutikno

PAGES 142 - 151

Paper 15: Constraint-Aware Automated Daily Meal Planning: A Controlled Comparison of Genetic Algorithm and Particle Swarm Optimization

Authors: Kohei Arai, Diva Kurnianingtyas, Nathan Daud

PAGES 152 - 159

Paper 16: Critical Vendor-Related Failure Factors in Outsourced ICT Projects: Insights from the Malaysian Public Sector

Authors: Masittah Musatapa, Fazlina Mohd Ali, Nurhizam Safie Mohd Satar, Azran Ahmad, Surya Sumarni Hussein

PAGES 160 - 168

Paper 17: Empowering Educators to Develop Augmented Reality Applications by Integrating Computational Thinking and Experiential Learning

Authors: Najwa Abd Ghafar, Nazatul Aini Abd Majid, Mohd Nor Akmal Khalid, Shinobu Hasegawa, Tan Siok Yee, Lam Meng Chun

PAGES 169 - 179

Paper 18: A Bidirectional Cross-Modal Attention Framework for Multimodal Sentiment Analysis

Authors: B S Harish, C K Roopa, M S Kendagannaswamy, T S Kaveri

PAGES 180 - 189

Paper 19: Artificial Intelligence for Rheumatoid Arthritis and Osteoporosis Diagnosis from Radiography, Computed Tomography, and Multimodal Clinical Data: A Systematic Review

Authors: Moulay Youssef ICHAHANE, Mohamed LACHGAR, Abdelouafi IKIDID, Ahmed DOURHRI, Mohamed EL GHAZOUANI, Noureddine ASSAD

PAGES 190 - 199

Paper 20: A Hybrid System with Automatic Classification and RAG for Moroccan Court Rulings

Authors: Sara IBRIHICH, Ahmed OUSSOUS, Ouafaa IBRIHICH, Mustapha ESGHIR

PAGES 200 - 209

Paper 21: IoT-Enabled Smart Cap for Real-Time Stress Monitoring

Authors: Bairavanathan Maha Sri, Kala Devi Managuran, Rajermani Thinakaran, Narayana Lokesh T, Punitha Shri M, Jeya Mithra K, Meenakshi B, Raja Subramanian R, Tong Mingjie

PAGES 210 - 218

Paper 22: A Hybrid Deep Learning Framework for Scientific Correction Text Classification

Authors: Garima Sharma, Vikas Tripathi, Vijay Singh

PAGES 219 - 229

Paper 23: AttendEase: Fraud-Resistant Attendance Tracking in Higher Education Using Dynamic Tokens and Geofencing

Authors: Silvia Gaftandzhieva, Gabriela Lyavova

PAGES 230 - 242

Paper 24: CMST-FL: Cross-Missingness Stable Trust with Evidence-Guided Preprocessing for Federated Healthcare Signal Imputation

Authors: Benachir Rigalma, Soumia Ziti, Meryam Belhiah, Hamid Ouhnni

PAGES 243 - 253

Paper 25: A Fairness–Utility Evaluation Framework for Assessing Large Language Models (LLMs)

Authors: Samah Alhazmi

PAGES 254 - 261

Paper 26: Accuracy, Stability, and Computational Efficiency of Deep CNN Architectures for Apple Leaf Disease Classification

Authors: Brahim Ouben Hssain, Khalil Ladrham, Noureddine El Barbri, Rachid El Ayachi, Abdellah Halimi

PAGES 262 - 272

Paper 27: Confidence-Gated Indonesian Grammatical Error Correction Under Synthetic-to-Authentic Domain Shift

Authors: Eko Suroso, Onok Yayang Pamungkas, Laily Nurlina, Sudiyono, Jimat Susilo, Eko Priyanto, Hera Septriana

PAGES 273 - 283

Paper 28: CompCap-Stego: Computable Capability-Pointer Steganography with Hidden Policy Evaluation and Verifiable External Payload Retrieval

Authors: Mohammad Othman Nassar

PAGES 284 - 299

Paper 29: A Machine Learning–Driven Information Systems Readiness Index for Predicting Strategic Cloud Adoption in Ghanaian Tertiary Education

Authors: Emmanuel Ofotsu Kwesi Bannor, S. Sarah Maidin, Vinayakumar Ravi, Nguyen Thi Thu Thuy, Nghiem Thi-Lich

PAGES 300 - 306

Paper 30: Color and Emotion in User Experience Design for Older Adults' Health Applications: A Multi-Database Bibliometric and Evidence-Mapping Study

Authors: YangFan He, Azizah Che Omar, Sobihatun Nur Abdul Salam

PAGES 307 - 318

Paper 31: Ablation-Based Climate-Aware Feature Engineering for Rice Productivity Prediction Using Multi-Source Agricultural Data

Authors: R. Hadapiningradja Kusumodestoni, Fatchul Arifin, Mutiara Nugraheni, Desti Setiyowati

PAGES 319 - 329

Paper 32: Deep Neural Network Architectures for Real-Time Drone Detection in Aerial Surveillance Systems: A Comparative Evaluation of YOLOv7, YOLOv8, and YOLOv11

Authors: Nazerke Abylay, Elvira Kadylbekkyzy, Bakhytzhan Kulambayev, Bayan Moldakalykova, Zharkyn Bimoldina, Saken Mambetov, Kuanysh Dossanbek

PAGES 330 - 340

Paper 33: Next-Gen Agriculture: Enhancing Animal Welfare Through IoT, Edge Devices, and Artificial Intelligence

Authors: Benlahsiniya Maroua, Ait Abdelouahid Rachida, Marzak Abdelaziz

PAGES 341 - 351

Paper 34: Predicting Multi-KPI Warehouse Dynamics: A Simulation-Based Evaluation of Fuzzy-Integrated Neural Architectures

Authors: Mohamed Amine Frikha

PAGES 352 - 358

Paper 35: Benchmarking ML-Based DDoS Detection Algorithms: Efficiency, Effectiveness, and Generalization

Authors: Zhilbert Tafa, Aurora Uka

PAGES 359 - 368

Paper 36: A Confidence-Calibrated Adaptive Ensemble Voting Framework for Stock Price Direction Prediction

Authors: Yuan Lukito, Nugroho Agus Haryono, Raden Gunawan Santosa

PAGES 369 - 383

Paper 37: Leveraging Cloud Computing Engineering for Sustainable Education: A Computational Evaluation Framework

Authors: Emmanuel Ofotsu Kwesi Bannor, S. Sarah Maidin, Vinayakumar Ravi, Nguyen Thi Thu Thuy, Nghiem Thi-Lich, Adade Sedom Percy

PAGES 384 - 394

Paper 38: An Energy-Efficient and Modeling Attack-Resilient SNN-Driven PUF Security for Internet of Medical Things

Authors: Agila Harshini T, Harini Sriraman

PAGES 395 - 402

Paper 39: A Machine Learning Framework for Wheat Seed Variety Classification Using GLCM-Based Texture Analysis and Feature Optimization

Authors: Muhammad Munawar Ahmed, Malik Muhammad Saad Missen, Hannan Adeel, Ahamad Zaki Bin Mohamed Noor, Muzamil Malik

PAGES 403 - 413

Paper 40: Perceptual Evaluation of Garment Drape Intensity in AI-Generated Garment Images

Authors: Noriaki Kuwahara, Yuta Yamamoto, Dongeun Choi, Kazunari Hirakoso, Takashi Satou

PAGES 414 - 422

Paper 41: SGRL: A Hybrid Semantic-Graph Representation Learning Framework for Software Vulnerability Detection

Authors: Cong Bui Van

PAGES 423 - 436

Paper 42: Cross-Cohort Reliability of Machine Learning for Breast Cancer Survival Prediction

Authors: NADIR Younes, RACHDI Mohamed, AMHAIMAR Lahcen, BAKHOUYI Abdellah, KHALIDI Abderrahim, AZOUAZI Mohamed

PAGES 437 - 448

Paper 43: Unmasking the Unknown: Feature-Orchestrated Image Synthesis for Deep Learning-Based Threat Detection

Authors: Sruthi C K, Harini Sriraman

PAGES 449 - 458

Paper 44: Arabic Text Simplification and Readability Assessment for Educational Applications: A Systematic Literature Review

Authors: Khalid Essaadani, Soumia Ziti

PAGES 459 - 468

Paper 45: An Adaptive AI-Driven Code Optimization Framework Using Machine Learning and Explainable AI for Scalable Software Systems

Authors: Kavya B K, S.Annamalai, A Suresh Kumar

PAGES 469 - 482

Paper 46: Enhanced Distributed Bandwidth Allocation Algorithm for Network Latency Reduction in IEEE 802.17 Rings

Authors: Fahd Alharbi

PAGES 483 - 488

Paper 47: Hybrid Resource Allocation Technique Based on Double Q Learning and Metaheuristic Algorithms in Optimizing the Allocation of Tasks in Cloud Computing

Authors: V. Mahalakshmi, S. Poornima

PAGES 489 - 499

Paper 48: Generative AI for Data-Informed Educational Decision-Making: A Systematic Review and Framework

Authors: Husna Hafiza Razami, Abtar Darshan Singh, Roslina Ibrahim, Ali Najeeb, Helio Augusto da Costa Xavier Mauquei

PAGES 500 - 510

Paper 49: Correlation-Aware Hybrid Feature Selection for Efficient Machine Learning-Based Intrusion Detection in Smart Home IoT Networks

Authors: Samar M. Alqhtani

PAGES 511 - 523

Paper 50: AI-Based Preprocessing and R-Squared Weighted Federated Aggregation for Robust Healthcare Signal Imputation

Authors: Benachir Rigalma, Hamid Ouhnni, Meryam Belhiah, Soumia Ziti

PAGES 524 - 532

Paper 51: Development and Simulation of a Hue-Based HMI Application for Vegetable Leaf Greenness Analysis

Authors: Djohar Syamsi, Mimin Muhaemin, Oviyanti Mulyani, Hanif Fakhrurroja, Puput Dani Prasetyo Adi

PAGES 533 - 541

Paper 52: An Architecture-Aware Flutter Code Generation Framework for Enforcing Clean Architecture and Behavioral Design Patterns

Authors: Desy Intan Permatasari, Adam Shidqul Aziz, Maulidan Bagus Afridian Rasyid, Ahmad Jarir At-Thobari, Umi Sa'adah, Nailussa'ada, Muhammad Fajrul Falah Subakti

PAGES 542 - 552

Paper 53: A Unified Agent-Based Simulation Framework for Epidemic and Wildfire Propagation in Complex Adaptive Systems

Authors: Natasha Stojkovikj, Aleksandra Stojanova Ilievska, Limonka Koceva Lazarova, Daniel Eftimov, Irena Eftimova

PAGES 553 - 562

Paper 54: Play-Centric Design and Usability Evaluation of a Block-Based Programming Serious Game for Low Vision Children

Authors: Huang Sheng, Chau Kien Tsong, Wan Ahmad Jaafar Wan Yahaya

PAGES 563 - 574

Paper 55: Hybridization of FA-YOLOv8 Model with Zebra Optimization Algorithm for Improving Traffic Light Detection and Tracking Under Low-Light Conditions

Authors: Adhwa Salemi, Muhammad Arif Mohamad

PAGES 575 - 579

Paper 56: Enhanced Explainable Multi-Agent Generative Artificial Intelligence Framework for Personalized Education

Authors: Hesham M. A. Abdullah, Elham Alzain

PAGES 580 - 604

Paper 57: Data Analytics Framework for Crop Yield Prediction Based on the Impact of Climate Using a Hybrid Model

Authors: Narinder Kaur, Vishal Gupta

PAGES 605 - 616

Paper 58: Student-Disjoint Cross-Module Benchmarking for Calibrated Early Warning in Online Learning

Authors: Mika Lim

PAGES 617 - 623

Paper 59: Dynamic Ant Colony Optimization–Driven Feature Selection for GRU-Based Network Intrusion Detection

Authors: Zhaosheng Yang

PAGES 624 - 633

Paper 60: Data-Driven Modeling and Optimization of Wind Turbine Performance

Authors: Naim Baftiu, Ana Atanasova, Enis Baftiu, Tatjana A. Pachemska, Egzon Baftiu, Vladimir Zdraveski, Petre Lameski

PAGES 634 - 640

Paper 61: ATLAS-FL: Active Timing-Adversary-Resilient Secure Aggregation for Gateway-Assisted IoT, IIoT, and IoMT Federated Intrusion Detection

Authors: Mohammad Othman Nassar

PAGES 641 - 662

Paper 62: Adaptive Virtual Reality Exposure Therapy Using AI-Driven Anxiety Prediction from Physiological and Behavioral Data

Authors: Fayaz Hussain G, P. Chenna Reddy

PAGES 663 - 669

Paper 63: Predictive RFID Tour Monitoring Dashboard Integrating Multimodal Machine Learning and Contextual Data

Authors: Heng Jin Koey, Siti Zuraidah Ibrahim, Ismahayati Adam, Megat Syahirul Amin Megat Ali, Rashidah Che Yob, Sugchai Tantiviwat, Sazwan Syafiq Mazlan

PAGES 670 - 679

Paper 64: Design and Formalization of a Distributed Intelligent Edge-Fog-Cloud Architecture for Industry 4.0 Systems

Authors: Hafsa AMALIK, Hicham MEDROMI

PAGES 680 - 691

Paper 65: A Vision-Based Error Compensation Method for Accurate Robotic Positioning in Automated Marking

Authors: Olzhas Olzhayev, Akhanseri Ikramov, Sayat Ibrayev

PAGES 692 - 699

Paper 66: Fair Graduate Employment Prediction for Chinese Universities: DML Assisted Stratified Panel XGBoost with Time Varying Loss

Authors: Wang Yanlai, Ting Tin Tin, Jia Jinlong, Zhang Jingren, Xu Lin

PAGES 700 - 714

Paper 67: Design and Validation of an Agentless Web Application for IT Infrastructure Operations Management

Authors: Carlos Otárola-Mermau, Alex Pacheco-Pumaleque

PAGES 715 - 724

Paper 68: The Impact of Gamified Escape Room Activities on Student Learning and Engagement in a Computer Database Course

Authors: Po Chan Chiu, King Kuok Kuok, Hamizan Sharbini, Mohammad bin Hossin, Chih How Bong, Noor Hazlini Borhan

PAGES 725 - 733

Paper 69: Fall Detection System Using Accelerometer and Gyroscope Sensors with Backpropagation Neural Networks

Authors: Helmy Fitriawan, Alby Dzaky Hadi Pamungkas, Ezza Ahmad Faturohman, Sri Purwiyanti, Anisa Ulya Darajat, Teddy Surya Gunawan

PAGES 734 - 740

Paper 70: Double Deep Q-Network-Based Smart Channel Hopping for Reactive Jamming Mitigation in IoT Networks: A Reliability and Energy-Aware Approach

Authors: Enos Letsoalo, Topside Mathonsi, Tshimangadzo Tshilongamulenzhe, Daniel du Duplesis

PAGES 741 - 748

Paper 71: Comparative Evaluation of Synthetic Data Generation Methods for Binary Classification Problems with Class Imbalance

Authors: Jhonatan Esquivel, Christian Humpiri, Jose Vega, Nemias Saboya

PAGES 749 - 764

Paper 72: A Deep Learning-Based Real-Time Surveillance Framework for Mitigating Herbivorous Animal Intrusions in Rice Nursery Fields

Authors: Muhammad Juman Jhatial, Riaz Ahmed Shaikh, Rafaqat Hussain Arain

PAGES 765 - 776

Paper 73: Interpretable Deep Convolutional Neural Network Framework for Online Reputation Management Using Customer Reviews and Social Media Feedback

Authors: Abdullah Saad AL-Malaise, Bandar A. Alrami Alghamdi, Mahmoud Ragab

PAGES 777 - 784

Paper 74: Design of a Mobile Application with a Hybrid Recommendation System for Connecting Andean Farmers and Urban Consumers in Peru

Authors: Jorge Mayta, Astrid Boronda, Juan Montes

PAGES 785 - 793

Paper 75: AI-Driven Personalized Learning Framework for Computing Fields Recommendations in Secondary and Pre-University Contexts

Authors: Anas Inuwa Usman, Shahida Sulaiman, Ruhaidah Samsudin

PAGES 794 - 808

Paper 76: An Enhanced Deep Learning Framework Integrating Adaptive Whale Optimization and Natural Language Processing for Mental Health Disorder Diagnosis

Authors: Nameer El.Emam, Issa Qabajeh, Mhammed Almakadmeh

PAGES 809 - 822

Paper 77: MedViT-MT: A Multi-Task Medical Vision Transformer for Glioblastoma Brain Analysis with Temporal Encoding and Clinical Data Fusion

Authors: Sana Cheema, Ghulam Gilanie, Muhammad Faheem Mushtaq, Rania Aboalela, Ali Daud

PAGES 823 - 837

Paper 78: Predicting Next-Trading-Session Stock Movements from Arabic Corporate Announcements: A Hybrid Transformer-Boosted Ensemble

Authors: Abdullah Almusned, Mohammad Mehedi Hassan, Muhammad Al-Qurishi, Bader Alkhamees

PAGES 838 - 847

Paper 79: Meta-Probabilistic Stacking for Eco-Driving Behavior Profiling

Authors: Md Mohibul Alam, Abdullateef Oluwagbemiga Balogun, Noreen Izza Arshad, Hussaini Mamman, Duc Minh Le, Van Dai Pham, Manh Toan Nguyen

PAGES 848 - 866

Paper 80: How Emergent Architectures Solve the Bird's Eyes View Shortcomings: A Systematic Review

Authors: Abdelaziz Sahbani, Rihem Sebai, Tarek Bejaoui, Hela Mahersia

PAGES 867 - 883

Paper 81: Enhancing Trustworthy Mining Trend Analytics Through Hybrid Semantic and Advertisement Detection Models

Authors: Mohamed Ayari, Khulud Alashjaee, Balsam Alzamam, Aryam Alshammari, Milaf Alshammari, Aadil Alshammari

PAGES 884 - 897

Paper 82: MedEdge-Conf: A Lightweight Confidence-Scoring Framework for Edge-Deployed LLMs in IoMT Health Monitoring

Authors: Fahd Koraiche, Rachid Dehbi, Amine Dehbi

PAGES 898 - 908

Paper 83: HTBM: A Hybrid-Transformer-Based Method with Double Deep Q-Learning and Reptile Meta-Learning for Dynamic MEC Task Offloading

Authors: Nouhaila Moussammi, Mohamed El Ghmary

PAGES 909 - 919

Paper 84: XraySafe-YOLO: Compact Scale-Aware Fusion and Contour-Motivated Attention for Recall-Oriented Detection of Occluded Prohibited Items in X-ray Images

Authors: Liu Liu, Wei Huang

PAGES 920 - 936

Paper 85: Individualizing an Agent-Based Flood Evacuation Model from Survey Data

Authors: Tatsuki Fukuda

PAGES 937 - 947

Paper 86: Small-Object Detection of Personal Protective Equipment in Underground Mine Environments Using an Improved YOLO26 Architecture

Authors: Taoufik Saidani, Refka Ghodhbani, Yahia Said

PAGES 948 - 960

Paper 87: A Hybrid NLP Framework for Topic Relevance and Advertisement Detection in Trending Hashtags on X (Twitter)

Authors: Aadil Alshammari, Khulud Alashjaee, Balsam Alzamam, Aryam Alshammari, Milaf Alshammari, Mohamed Ayari

PAGES 961 - 968

Paper 88: Enhanced Harris Hawks Optimization Algorithm with Integrated Tabu Search for 3D Bin Packing Problems: Application in Logistics Industries

Authors: Yasser M. Ayid, Mohamed Meselhy Eltoukhy, Alaa Mokhtar, Rabie Mosaad

PAGES 969 - 979

Paper 89: Energy Personalized Recommender for the PowerEye Home Energy Management System

Authors: Reem Hejazi, Sofianiza Abd Malik, Anees Ara, Layla Alfawzan

PAGES 980 - 992

Paper 90: Anatomia: An Interactive 3D Visualization Tool for Enhancing Medical Anatomy Education

Authors: Iranga Mudalige, Hiruna Jayasuriya, Yeshan Gunawardane, Hashen Cooray, Imashi Dissanayake, Damitha Sandaruwan, Chamath Keppitiyagama, Mahappuge Shasika Eranda Karunadasa, Tony Mahadevan

PAGES 993 - 1007

Paper 91: Does It Generalize? A Cross-Dataset Study of Graph-Based Insider Threat Detection Beyond CERT

Authors: QASIM Mohamed Muhanna Alriyami, Mohd Murtadha Bin Mohamad

PAGES 1008 - 1019

Paper 92: Empirical Evaluation of Language Models for Arabic Financial Sentiment Analysis: A Comparative Study of Pretrained and General Large Language Models

Authors: Abdullah Almusned, Mohammad Mehedi Hassan, Muhammad Al-Qurishi, Bader Alkhamees

PAGES 1020 - 1032

Paper 93: Adaptive Differential Privacy in Federated Learning for Privacy-Preserving ICU Clinical Decision Support at the Edge

Authors: Moaad Almania, Anazida Zainal, Nurfazrina Binti Mohd Zamry, Muhammad Haris, Ahmad Alnawasrah, Fuad A. Ghaleb

PAGES 1033 - 1042

Paper 94: An Efficient and Reliability-Oriented Deep Learning Framework for Alzheimer's MRI Classification

Authors: Yashwant A, Rajaprakash S

PAGES 1043 - 1053

Paper 95: Multi-Level Contextual Semantic Modeling for Implicit Hate Speech Recognition

Authors: Rajkumar Panchal, Megha Jonnalagedda

PAGES 1054 - 1066

Paper 96: TrustEdge-AI: Uncertainty-Aware Dynamic Trust Intelligence for Adaptive Cyber Defense in IoT Edge Networks

Authors: Mohamed Ayari, Ahmed Alhomoud, Atef Gharbi, Yamen El Touati, Zeineb Klai

PAGES 1067 - 1075

Paper 97: FPGA-SoC Implementation of 3D-HEVC Encoder Based on MD-CNN

Authors: Nacir Omran, Amna Maraoui, Imen Werda, Hamdi Belgacem

PAGES 1076 - 1088

Paper 98: The Coordination Complexity Index: A Parameter-Free Checkpoint Diagnostic for Silent Coordination Failure in Cooperative Multi-Agent Reinforcement Learning

Authors: Chatchitsanu Pothisakha, Pudsadee Boonrawd, Siranee Nuchitprasitchai

PAGES 1089 - 1100

Paper 99: Predicting Crowd Agitation from Raw Video: A Lightweight Multimodal Regression Framework for Social Tension Estimation

Authors: Abderrahim Ouza, Khalid Ounachad, Mohamed El Ghmary, Ali Choukri

PAGES 1101 - 1113

Paper 100: Quantum Inspired Hybrid Machine Learning Framework Using Support Vector Machine for Financial Time Series Forecasting

Authors: Visalakshi Palaniappan, Iskandar Ishak, Hamidah Ibrahim, Fatimah Sidi, Mohamad Yusnisyahmi Yusof

PAGES 1114 - 1126

Paper 101: Eliminating Reference Mixing in Automotive Manufacturing: A DMAIC and Machine Vision Integration Approach

Authors: Morad Wafi, Mouad Danane, Abderrahim Bouzid, Abdelmajid El Ouadi

PAGES 1127 - 1138

Paper 102: Hodge-Guided Active Preference Elicitation for Efficient Pairwise Ranking

Authors: Eskander Bejaoui, Mohamed Ould-Elhassen Aoueileyine, Ridha Bouallegue

PAGES 1139 - 1149

Paper 103: EfficientNetB0-Based DeepFake Image Detection: A Transfer-Learning Framework for Robust and Deployable Media Verification

Authors: Marwa Hamza, Mohammad Abdallah

PAGES 1150 - 1155

Paper 104: N-DQN-RRT: A Hybrid Deep Q-Network Guided RRT Algorithm for an Efficient Path Planning in Two-Dimensional Environments

Authors: Wai Yan Naing Win, K. Kanagalakshmi, K. Priyadharshini, Zeyad A. T Ahmed, R. RajiniGanth, P. Ganesh Kumar, Atif Mahmood

PAGES 1156 - 1165

Paper 105: Deep Learning and Computer Vision-Based Modeling for a Real-Time Traffic Congestion Monitoring Framework

Authors: Neil Dustin Benedict A. Agner, Angelo B. Dela Cruz, Aster Benedict A. Mangabat, Raphael B. Alampay, Patricia R. Angela Abu

PAGES 1166 - 1174

Paper 106: Beyond the Login Screen: Teaching Full-Stack Defense-in-Depth from Software to Hardware Logic

Authors: Md Mufassal Ahmad, Mary Nusrat, Gahangir Hossain

PAGES 1175 - 1183

Adversarial Asymmetric Agents for Network Intrusion Detection Using Deep Reinforcement Learning

Curtis Rookard
Department of Computer Science and Technology, SUNY Empire State University, Saratoga Springs, NY 12866, USA

***Abstract*—Machine learning-based network intrusion detection systems (NIDS) increasingly operate in environments where adversaries can adapt their behavior in response to deployed defenses. However, most empirical IDS studies evaluate defenders against static, dataset-defined attack distributions and do not explicitly model an adaptive attacker with a distinct learning strategy or action space. This study proposes an adversarial asymmetric learning framework for intrusion detection in which an attacker agent dynamically selects network attacks while a defender agent classifies observed network flows. We evaluate three attacker–defender pairings derived from supervised deep neural network (DNN) and Deep Q-Network (DQN) agent types. We report accuracy, precision, recall, F1-score, and G-Mean under stratified 10-fold cross validation across traditional local area network, industrial control system (ICS)/supervisory control and data acquisition (SCADA), and internet-of-things (IoT) networking environments. Empirically, the network-flow classifier performs best when paired with an intelligent attacker using reinforcement learning, even when the defender itself is not reinforcement-learning based. Overall, the results show that learning asymmetry and multi-agent interaction can substantially alter measured detection outcomes and that supervised defenders remain highly competitive in the presence of intelligent adversarial agents.**

Keywords*—Network intrusion detection; cybersecurity; deep reinforcement learning; deep q-networks; adversarial machine learning; multi-agent learning

I. Introduction

The continued growth of information technology and information processing systems worldwide have seen an increase of computing related resources. Underpinning the development and expansion of these information systems is the growth and proliferation of computer networks which now have to accommodate more than just traditional computing platforms. New technologies, such as internet-of-things, embedded systems, and industrial control systems and supervisory control and data acquisition platforms now comprise a significant share of all computing systems [1]. Application of these systems necessitate the need for reliable, trustworthy network connectivity. Unfortunately, the continued expansion and adoption of these computer systems have introduced additional attack vectors for adversaries to exploit.

Pinnacle to the protection of these computing systems is a need for robust cybersecurity technologies to ensure the threats do not comprise the aforementioned information systems. In addition to traditional cybersecurity techniques such as virtual local area networks/network segmentation, access controls, and intrusion prevention and detection systems play a crucial role [2]. Most intrusion detection systems today rely on robust machine learning mechanisms to ascertain whether to permit or to deny traffic from entering a computer network or system [3].

With the advent of more sophisticated artificial intelligence techniques, such as large language models [4], and generative adversarial networks (GANs) [5], there has been a renewed interest in applying other methods for cybersecurity, specifically for intrusion detection. As such, we have seen an increase in the utilization of more advanced machine learning algorithms, extending into the reinforcement learning domain. In reinforcement learning, an intelligent agent interacts with environmental stimuli in order to obtain the highest reward possible. In cases relating to cybersecurity, this could be presented in the sense of a cybersecurity agent interacting with the networking environment.

Interestingly, with the increased utilization of deep reinforcement learning agents for defense, there has also been a rise in the usage of deep reinforcement learning agents for offensive security purposes [6]. The combination of defender and attacker learning introduces a methodological challenge for intrusion detection evaluation: the defender is no longer tested only against a static distribution of attacks, but against an opponent whose behavior can change the mixture of observed traffic. This motivates a framework that can compare symmetric and asymmetric attacker–defender learning configurations under a common experimental protocol.

In this work, we make the following contributions to the intrusion detection literature:

- We propose an adversarial asymmetric learning framework for network intrusion detection in which the attacker and defender are modeled as interacting agents with potentially different learning mechanisms and action spaces.
- We examine the role-dependent effect of learning architecture by comparing a symmetric DNN–DNN baseline against two role-reversed asymmetric configurations: a DQN attacker with a DNN defender and a DNN attacker with a DQN defender. This controlled design allows us to determine whether the effect of reinforcement learning depends on the role in which it is deployed.

- We evaluate these scenarios on three widely used intrusion detection datasets spanning distinct network types: local area networking represented by the CIC-IDS 2017 dataset [7], industrial control system (ICS) and supervisory control and data acquisition (SCADA) environments represented by the DNP3 dataset [8], and IoT networking represented by the ACI-IoT dataset [9].
- We conduct a comprehensive comparative analysis across scenarios and datasets using standard intrusion detection metrics of accuracy, precision, recall, F1-score, and G-Mean to quantify how asymmetric learning changes measured detection performance under adaptive adversarial interaction.

Based on the literature reviewed in Section II, we identify a gap in the systematic evaluation of the directional placement of supervised and reinforcement learning mechanisms across opposing attacker and defender roles. The novelty of our work therefore lies not in the use of DNN or DQN models individually, but in the controlled comparison of their role asymmetry within an explicit attacker–defender interaction in which the attacker can influence the mixture of network traffic encountered during learning. The rest of our study is described in the following sections. Section II provides a greater overview of the related literature with both intrusion detection and reinforcement learning. Section III covers the methodology presented in our manuscript, while section IV provides the numerical analysis of our results. Section V provides the discussion of our work, and finally section VI concludes our manuscript.

II. Related Literature

After a careful review of literature, we find a lack of prior research examining deep reinforcement learning for intrusion detection under the same premises we adopt in our work. We aim to most notably present a competitive formulation with asymmetric agents using a combination of both deep reinforcement learning agents and traditional deep learning agents for both offensive and defensive intrusion detection. While numerous studies evaluate conventional machine learning and deep learning classification, and a growing body of work applies reinforcement learning to intrusion detection and traffic classification, these approaches typically assume a single decision-maker, that is, the defender, and do not explicitly model an adaptive adversary with a different learning strategy, information advantage, or action space. Our proposed framework is based upon this asymmetry, enabling a more realistic characterization of attacker–defender dynamics across heterogeneous network scenarios.

Accordingly, this section summarizes the most representative related works that inform our experimental design and baselines. We first review established intrusion detection results on widely used datasets to situate performance expectations under common supervised-learning setups, then discuss the most relevant intelligent intrusion detection studies utilizing reinforcement learning in the broader literature. Finally, we highlight similar works applying adversarial attacks for cybersecurity to thus better position our work against the current literature.

A. Traditional Intrusion Detection

Intrusion detection systems (IDSs) are commonly categorized by both deployment scope, that is either host-based vs. network-based, and detection philosophy, which are divided into signature-based vs. anomaly-based. Signature-based systems match traffic to known patterns and typically offer low false-alarm rates for previously observed attacks, whereas anomaly-based systems aim to detect deviations from a learned notion of "normal" behavior and are therefore better positioned to identify novel or low-footprint threats. As operational networks have diversified through the additional of new technologies and software, anomaly-based IDS systems have increasingly relied on statistical learning models that map flow-based or packet-derived features to either benign or attack labels. Broad surveys in the cyber analytics literature document this progression and show that classical methods such as decision trees, support vector machines, k-nearest neighbors, and shallow ensembles remain competitive in some environments, but also that their performance is often sensitive to feature engineering choices, dataset artifacts, and class imbalance [10].

More recent work has accelerated the shift from manual feature engineering toward deep learning (DL) approaches that can learn nonlinear decision boundaries and capture higher-order feature interactions. Deep neural network (DNN)-based IDS models are typically trained in a supervised manner on labeled flow records, and they have been extended with temporal architectures, including recurrent neural networks (RNNs), gated recurrent units (GRUs), and long-short term memory (LSTM) networks alongside representation learning to better model sequential dependencies or reduce reliance on handcrafted preprocessing. For example, Vinayakumar et al. evaluate DNN-based intrusion detection across multiple benchmark datasets and highlight that modern network behavior and rapidly evolving attack patterns motivate continual dataset refresh and careful cross-dataset benchmarking [11]. At the same time, end-to-end NIDS research has explored learned representations that incorporate contrastive objectives and hierarchical feature extraction to improve generalization under distribution shift, which is especially relevant when models trained on one environment are deployed in another [12].

Because IDS evaluation is tightly coupled to dataset construction, a substantial body of work focuses on the limitations of legacy benchmarks and on the design of more realistic datasets. In the work of Tavallaee et al. these researchers present inherent issues in KDD Cup 1999 dataset. These issues include redundancy and biased difficulty distributions, and they thus propose the NSL-KDD dataset to mitigate those artifacts, which partly explains why many "high-accuracy" results on older benchmarks fail to translate to operational settings [13]. Dataset-centric surveys further show that modern datasets attempt to better reflect contemporary traffic mixtures, feature sets, and attack diversity, while still exhibiting challenges such as severe imbalance and dataset-specific leakage [14], [15]. For instance, the UNSW-NB15 was created to include a hybrid of modern normal activity and synthesized contemporary attacks, thus improving upon the previous NSL-KDD dataset [16]. Similarly, the CIC-IDS 2017 dataset provides labeled flow records derived from network packet captures and has become

a standard evaluation dataset for supervised NIDS baselines [7]. These findings motivate our use of multiple datasets spanning heterogeneous network types (LAN, ICS/SCADA, and IoT) and allows us to better ascertain IDS performance.

B. Reinforcement Learning for Intrusion Detection

Reinforcement learning has been increasingly applied to cybersecurity because it often mathematically models adaptive decision-making under uncertainty, and partial observability. In many cyber defense formulations, a reinforcement learning agent learns a policy for sequential actions such as dynamic access control, moving target defense, deception, or response orchestration. In contrast to static supervised learning, reinforcement learning can optimize long-run objectives by explicitly learning from interaction feedback rather than relying solely on fixed labels.

Within the intrusion detection literature specifically, reinforcement learning appears in two primary strands. The first treats intrusion detection as a sequential classification problem where the agent's action is the predicted class and the reward encodes correctness or utility; Deep Q-Learning variants have been proposed to learn intrusion detection policies under this framing [17]. The second strand uses reinforcement learning as an optimization layer for IDS pipelines. These pipelines aim to select features, tune thresholds, or adapt detection policies online in non-stationary environments. Ren et al. propose a deep reinforcement learning-based intrusion detection method that emphasizes feature selection which provides performance benefits on modern traffic datasets [18]. More recent applied studies continue to report that deep reinforcement learning can be competitive with supervised baselines under certain conditions, especially in situations wherein the environment is dynamic or when a defender must trade off sensitivity with operational costs [19], [20].

However, a consistent limitation in much of the intrusion detection literature applying reinforcement learning is that the learning environment is often derived from a fixed dataset stream, and the "adversary" is modeled implicitly through pre-labeled attack samples rather than as an adaptive opponent. Surveys of deep reinforcement learning for intrusion detection, such as those seen in IoT contexts, note that many published approaches assume a single decision-maker, often the defender, and do not explicitly study the stability issues that arise when both attacker and defender learn simultaneously [21]–[24]. Instances incorporating both attacker and defender interactions are more naturally captured by stochastic or Markov games to better learn robust strategies against adaptive opponents [25]–[27]. Our work directly targets this gap for intrusion detection by evaluating asymmetric and symmetric pairings for both attacker and defender across heterogeneous network settings, thereby empirically characterizing how learning asymmetry alters detection outcomes.

C. Adversarial Methods for Cybersecurity

Adversarial machine learning has become a central concern for ML-based IDS because attackers may attempt to evade detection through evasion attacks, corrupt training data through data poisoning, or extract model behavior such as through model inference attacks, to enable more effective intrusions. In NIDS, these threats are complicated by domain constraints; feature perturbations must correspond to realizable network traffic, preserve protocol validity, and often maintain malicious functionality. Recent surveys propose taxonomies tailored to NIDS and synthesize both attack and defense mechanisms, emphasizing that robustness claims in cybersecurity must account for traffic realism and threat-model fidelity [28]–[30].

A growing set of works operationalizes these concerns by designing attacks that respect practical constraints. Sheatsley et al. show that even with a constrained attack surface, NIDS models can remain vulnerable to adversarial examples and that designing realistic networks alone does not guarantee robustness against adversarial attacks [31]. GAN-based approaches have also been explored to generate adversarial traffic flows. Zolbayar et al. introduce NIDSGAN to craft practical adversarial flows and evaluate their ability to circumvent ML-based NIDS under realistic feature constraints [32]. Similarly, Debicha et al. propose Adv-Bot, a black-box evasion method that aims to produce botnet traffic capable of evading detection while still preserving intended malicious behavior, and they also discuss mitigation strategies [33]. Rookard shows that even deep learning-based NIDS can still be heavily affected by adversarial attacks on training data [34]. Complementary lines of work study defender strategies under limited query access and partial feedback, highlighting the challenges defenders face when they cannot directly observe model outputs or when probing risks triggering detection [35].

This adversarial literature motivates two design choices in our work. First, it reinforces the need to evaluate IDS methods under conditions that reflect adaptivity and strategic behavior rather than assuming static attack distributions. Second, it suggests that robustness should be assessed not only by aggregate accuracy but by sensitivity-focused metrics and by stability across datasets. Rather than generating adversarial perturbations in feature space, our work models adaptivity through an explicit attacker agent whose learning strategy may differ from the defender's learning, providing a complementary, game-aware perspective on the adversarial presence in network intrusion detection.

III. Methodology

In this section, we cover the primary methodologies used in our research, specifically, we cover Markov decision processes, and their usage for deep Q-networks. We then discuss our algorithm and datasets applied in our work.

A. Markov Decision Process

We formulate our intrusion detection problem as a Markov decision process (MDP), a standard framework for modeling sequential decision-making in stochastic environments [36], [37]. An MDP is typically defined by the tuple $\langle S, A, T, R, \gamma \rangle$, where S is the set of states, A is the set of actions, $T(s' \mid s, a)$ denotes transition dynamics of moving to new state s' given action a and current state s, $R(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor that controls the weighting of future rewards.

Accordingly, we model network intrusion detection as follows: let S denote the set of network observations in the intrusion detection dataset, where each state corresponds

to a single network flow represented by a feature vector; these features include flow duration, source bytes, destination bytes, protocol type, and related statistics. The action space A corresponds to the defender agent's classification decision for a given flow, either being benign or an attack. The transition function T captures how traffic evolves from one observed flow to the next, reflecting the underlying mixture of benign and adversarial behavior present in the network stream. Finally, the reward signal is defined as a binary outcome: the agent receives $r = 1$ for a correct classification and $r = 0$ otherwise, i.e., $r \in \{0, 1\}$. The MDP formulation is therefore used as a simplified, dataset-driven model of attacker–defender interaction using the network flow traffic data.

B. Deep Q-Networks

We solve the above MDP using value-based reinforcement learning. In particular, we build on the Q-learning algorithm [37], which learns an action-value function $Q(s, a)$ representing the expected discounted return gained by our intelligent agent taking action a in state s through following an optimal policy. This optimal action-value function thus satisfies the Bellman optimality relationship,

$$Q^*(s, a) = \mathbb{E}\Big[r + \gamma \max_{a' \in A} Q^*(s', a') \,\Big|\, s, a\Big], \tag{1}$$

and Q-learning iteratively approximates Q^* via temporal-difference updates of the form

$$Q(s, a) \leftarrow Q(s, a) + \alpha\Big(r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)\Big), \tag{2}$$

where, α is the learning rate.

We therefore employ a Deep Q-Network (DQN), which approximates $Q(s, a)$ using a neural network parameterized by θ, denoted $Q(s, a; \theta)$ [38] as network-flow feature spaces are high-dimensional and continuous and maintaining a tabular Q-function is infeasible. Given a state feature vector, represented as a network flow, as input, the network outputs a vector of Q-values, one per defender action which thus allows the agent to select actions using an ϵ-greedy policy (i.e., choosing $\arg\max_a Q(s, a; \theta)$ with probability $1-\epsilon$ and exploring otherwise). To stabilize learning, we use experience replay and a target network: transitions (s, a, r, s') are stored in a replay buffer and sampled uniformly to break temporal correlations, while a separate set of parameters θ^- is periodically updated to provide a slowly moving target. The DQN is trained using the Huber temporal-difference loss between the predicted action value, $Q(s, a; \theta)$ and the target $y = r + \gamma \max_{a' \in A} Q(s', a'; \theta^-)$. The Huber loss reduces sensitivity to large temporal-difference errors while retaining a quadratic penalty for smaller errors.

C. Adversarial Asymmetric Agents for Reinforcement Learning

To capture adaptive attacker–defender dynamics that are not represented in standard supervised IDS evaluation, we extend the single-agent MDP formulation in Sections III-A–III-B to a two-agent competitive learning environment. Rather than assuming that malicious traffic arrives according to a fixed distribution, as in offline classification, we model an attacker agent that actively influences which type of traffic is encountered by the defender. This design allows us to study how detection performance changes when learning is symmetric, wherein both agents use the same class of learner, versus asymmetric wherein agents use different learning paradigms, which is central to the comparative scenarios evaluated in this work.

1) Two-agent game formulation: We model the interaction as a two-player stochastic game, akin to a Markov game), defined by the tuple $\langle S, A_D, A_A, T, R_D, R_A, \gamma \rangle$, where S is the shared state space, A_D and A_A are the defender and attacker action spaces, T defines the transition dynamics, R_D and R_A are the respective reward functions, and $\gamma \in [0, 1)$ is the discount factor. The state $s_t \in S$ corresponds to the observed feature vector of a network flow at time step t. The defender action $a_t^D \in A_D$ is the predicted class label, while the attacker action $a_t^A \in A_A$ represents the attacker's selection of an attack mode that determines the type of traffic the defender will face next. This reflects a realistic adversarial capability: while a defender may only observe a flow and decide how to label it, an attacker can adaptively choose which tactics/techniques to employ over time.

2) Reward structure and competitive objective: We use a competitive (zero-sum) reward design that directly encodes detection success. Let $\mathbb{I}[\cdot]$ denote the indicator function and let y_t be the ground-truth label (mapped to the defender's label space when coarse categories are used). The defender receives:

$$r_t^D = \mathbb{I}\big[a_t^D = y_t\big], \tag{3}$$

and the attacker receives the complementary reward

$$r_t^A = 1 - r_t^D. \tag{4}$$

Thus, the defender maximizes correct classification over time, while the attacker maximizes defender misclassification. Thus, attacker reward reflects defender error induced through adaptive traffic-mixture selection. Because the attacker neither perturbs flow features nor synthesizes adversarial examples, this advantage represents a mixture-selection effect rather than conventional feature-space evasion

3) Asymmetry in action spaces and information: Our framework is intentionally asymmetric in two ways. First, the attacker and defender may have different action spaces: the defender acts in a coarse IDS label space, e.g. denial-of-service, while the attacker acts in a fine-grained space that captures specific attack types, e.g. GoldenEye and Slowloris attacks as seen in the CIC-IDS 2017 dataset. Second, the attacker controls the traffic mixture by selecting which class of flows is drawn next, whereas the defender reacts to whatever flow is observed. These design choices jointly model a realistic attacker advantage: even if the defender's classifier is strong on average, an adaptive attacker can concentrate effort on classes wherein the defender is weakest. Before selecting an action, the attacker observes the current normalized flow-state representation. The attacker does not directly observe the defender's model parameters, class probabilities, or an explicit history of defender decisions, instead adapting from the scalar

reward and replayed interaction history, whereas the supervised DNN attacker is trained using labeled data and subsequently acts as a fixed selector. The two attackers, the DNN attacker and the DQN attacker, share the same action space but do not optimize equivalent learning objectives.

4) Agent instantiations and the three evaluated pairings: We implement each agent as either 1) a standard deep neural network (DNN) classifier trained via supervised learning, or 2) a Deep Q-Network (DQN) trained via reinforcement learning. This yields the three scenarios evaluated in Section I: 1) *Both DNNs*, where the attacker follows a fixed DNN-based policy for selecting attack modes and the defender follows a fixed DNN classifier; 2) *DNN Defender*, where the defender is supervised and fixed while the attacker learns a DQN policy; 3) *DQN Defender*, where the attacker is supervised and fixed while the defender learns a DQN. For DNN agents, the policy is deterministic at inference time, whereas DQN agents act using an ϵ-greedy strategy during training to balance exploration and exploitation.

5) Learning procedure for DQN agents: When an agent is instantiated as a DQN, it learns an action-value approximation $Q(s, a; \theta)$ for its respective action space (attacker or defender). Each agent maintains its own replay buffer and target network and minimizes the temporal-difference loss described in Section III-B. To improve stability, we train with standard DQN stabilizers (experience replay and target networks) and update each agent from its own buffer using mini-batches drawn from recent interaction experience. Empirically, this setup is sufficient for stable learning in our dataset-driven environments and enables direct comparison to supervised baselines under identical state representations and evaluation metrics.

6) Training and testing process within cross-validation: Within each cross-validation fold, supervised DNN agents are trained only on the training split. DQN agents interact only with flows drawn from the training split while learning through the aforementioned sampling mechanism. After training, we evaluate the defender's detection performance on the held-out fold using the same metrics reported in Section IV-A. This ensures that the reported results measure generalization to unseen flows while still allowing attacker adaptivity during training to shape the defender's learning experience.

D. Datasets

In this section, we cover the intrusion detection datasets used across our three intrusion detection environments.

1) CIC-IDS 2017 dataset: The University of New Brunswick's Canadian Institute for Cybersecurity crafted the CIC-IDS 2017 dataset to address limitations in present intrusion detection datasets. Improving upon the previously released NSL-KDD dataset [13] and UNSW-NB15 datasets [16], this dataset better resembles real world network traffic present in a local area network. Traffic used to create the CIC-IDS 2017 dataset originated from network packet capture (.pcap) files taken over a five-day period.

In this dataset, there are over 2.8 million instances of traffic, with several instances being more reflective of normal, benign traffic. There are 78 features present in the dataset which include network flow characteristics such as the duration of the network flow, the total length of the packet in bytes, ports used in communication, transmission control protocol flags (TCP) present. An overview of the distribution of the network traffic is provided in Fig. 1.

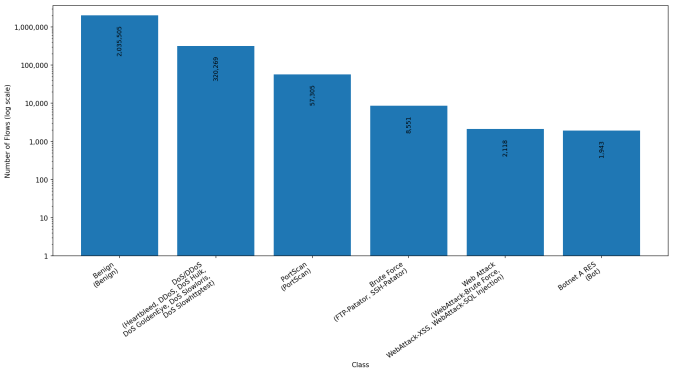


Fig. 1. Distribution of traffic present in the CIC-IDS 2017 dataset using a logarithmic scale for the number of network flows. Traffic class raw counts are indicated on each bar. Specific attacks for each traffic class are indicated in parentheses.

2) DNP3 dataset: The DNP3 intrusion detection dataset, created by the Internet of Things and Applications (ITHACA) lab at the University of Western Macedonia, provides a means to determine the efficacy of machine learning and deep learning models on industrial control systems and supervisory control and data acquisition (SCADA) platforms. The dataset is centered upon the Distributed Networking Protocol version 3, which is a common protocol used in many SCADA environments [8].

Across both the training and testing datasets on the DNP3 intrusion detection dataset, there are over seven thousand unique flows with 84 different features reflective of netflow traffic. We opt to utilize the balanced training and testing datasets as provided by the ITHACA lab for our work.

3) ACI-IoT dataset: In order to more accurately assess threats, especially within IoT networking environments, the Army Cyber Institute at the United States Military Academy developed a novel dataset, the ACI-IoT Network Traffic dataset [9] which captures the complexities of traffic originating from various IoT devices with both wired and wireless connections to the network.

There are over 1.2 million instances of unique traffic present in the ACI-IoT dataset, with over 84 different features, akin to the features captured in the netflow traffic of the CIC-IDS 2017 dataset. An overview of the distribution of the network traffic is provided in Fig. 2.

E. Proposed Experimental Design

This subsection specifies the complete experimental protocol used to evaluate adversarial asymmetric learning in intrusion detection. Our design is driven by two requirements: 1) the comparison across the three attacker–defender pairings must be fair (identical state representation, identical train/test splits, and consistent evaluation metrics), and 2) the interaction between attacker and defender must be *adaptive* for

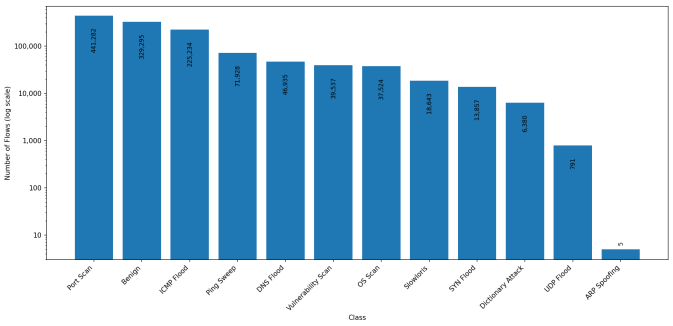


Fig. 2. Distribution of traffic present in the ACI-IoT 2023 dataset using a logarithmic scale for the number of network flows. Traffic class raw counts are indicated on each bar.

reinforcement learning agents while still permitting out-of-sample generalization measurement using cross validation.

1) Preprocessing and state representation: For each dataset, we construct a tabular feature matrix where each row corresponds to a single network flow and each column corresponds to a numeric flow feature. Categorical fields are converted to numeric form using standard encodings, and missing/infinite values are removed from the dataset. To ensure comparable optimization dynamics across datasets and to stabilize both DNN and DQN training, features are scaled using statistics computed on the training split only, and the same transformation is applied to the corresponding test fold. The resulting normalized feature vector constitutes the environment state $s_t \in S$ for both agents.

2) Label spaces and coarse–fine mapping: Each dataset includes a potentially large set of specific attack labels. For the defender, we define a *coarse* action space A_D consisting of benign traffic plus a small set of broad attack categories appropriate for the dataset. For the attacker, we define a *fine* action space A_A that corresponds to the available attack subclass labels in the dataset. A deterministic mapping $g(\cdot)$ maps each fine label to the defender's coarse label space, and correctness is computed using the coarse label $y_t^{(coarse)} = g(y_t^{(fine)})$.

a) Cross-validation process: All reported results are obtained using stratified 10-fold cross validation. In each fold, we split data into a training portion and a held-out test portion. Any preprocessing parameters and any supervised model fitting are performed strictly on the training split. Reinforcement learning agents interact only with the training split during learning, and the learned defender policy is then evaluated on the held-out test fold. This protocol ensures that attacker adaptivity can shape training-time exposure, while the final metrics quantify generalization to unseen flows.

3) Dataset-driven interaction environment: Within each fold, we construct an episodic environment from the training split. At each time step t, the attacker selects an action $a_t^A \in A_A$, which indexes a subset (pool) of training flows whose fine-grained label corresponds to a_t^A. The environment then samples the next observation x_{t+1} from that pool (uniformly at random). The defender observes the resulting state $s_{t+1} = x_{t+1}$ and selects an action $a_{t+1}^D \in A_D$ corresponding to its predicted coarse label. The defender reward is binary,

$$r_{t+1}^D = \mathbb{I}\big[a_{t+1}^D = g(y_{t+1}^{(fine)})\big], \tag{5}$$

and the attacker reward is complementary,

$$r_{t+1}^A = 1 - r_{t+1}^D. \tag{6}$$

Episodes terminate after a fixed horizon or when insufficient samples remain in one or more pools, depending on the dataset. To reduce sampling artifacts, we attempt to avoid immediate reuse of the same flow within an episode, while permitting reuse across episodes to ensure adequate training exposure.

4) Agent configurations and the three scenarios: We evaluate the following attacker–defender pairings under identical folds and preprocessing:

- Both DNNs: Both attacker and defender are trained as supervised DNN classifiers and then fixed during interaction/evaluation.
- DNN Defender: The defender is a supervised DNN classifier (fixed after training) and the attacker learns an adaptive policy with a DQN.
- DQN Defender: The attacker is a supervised DNN selector (fixed after training) and the defender learns a DQN policy for classification.

In supervised configurations, the attacker's DNN acts as a selector over A_A, choosing the specific attack, and the defender's DNN acts as a classifier over A_D, choosing the type of attack. In reinforcement learning configurations, the DQN output dimension matches the corresponding agent action space. Our experimental design is described in Fig. 3.

5) Supervised learning baselines (DNN): For DNN agents, we train a feed-forward multilayer perceptron on the training split using categorical cross-entropy loss and the Adam optimizer. We employ standard regularization and early stopping based on a validation subset carved from the training fold to prevent overfitting. After training, the resulting model is used as a deterministic policy, with actions selected using argmax over predicted class probabilities, during interaction and test-fold evaluation.

6) Reinforcement learning agents (DQN): Reinforcement learning agents (DQN). For DQN agents, each agent maintains an online Q-network, a target network, and an experience replay buffer. The DQN uses the Adam optimizer with a learning rate of 1×10^{-4}, with the discount factor, γ = 0.001, a replay-buffer capacity of 5,000 transitions, and replay mini-batches of 256 samples. The Q-network uses 128 hidden units with ReLU activation, and the target network is synchronized every 100 optimization updates. Exploration follows an ϵ-greedy policy initialized at ϵ = 1.0, with a minimum value of 0.10 and decay coefficient of 0.995 applied at 100-action intervals. Each episode contains 100 interaction steps. The attacker-DQN configuration was trained for 50 episodes (5,000 interaction steps), and the defender-DQN configuration was trained for 50 episodes (5,000 interaction steps).

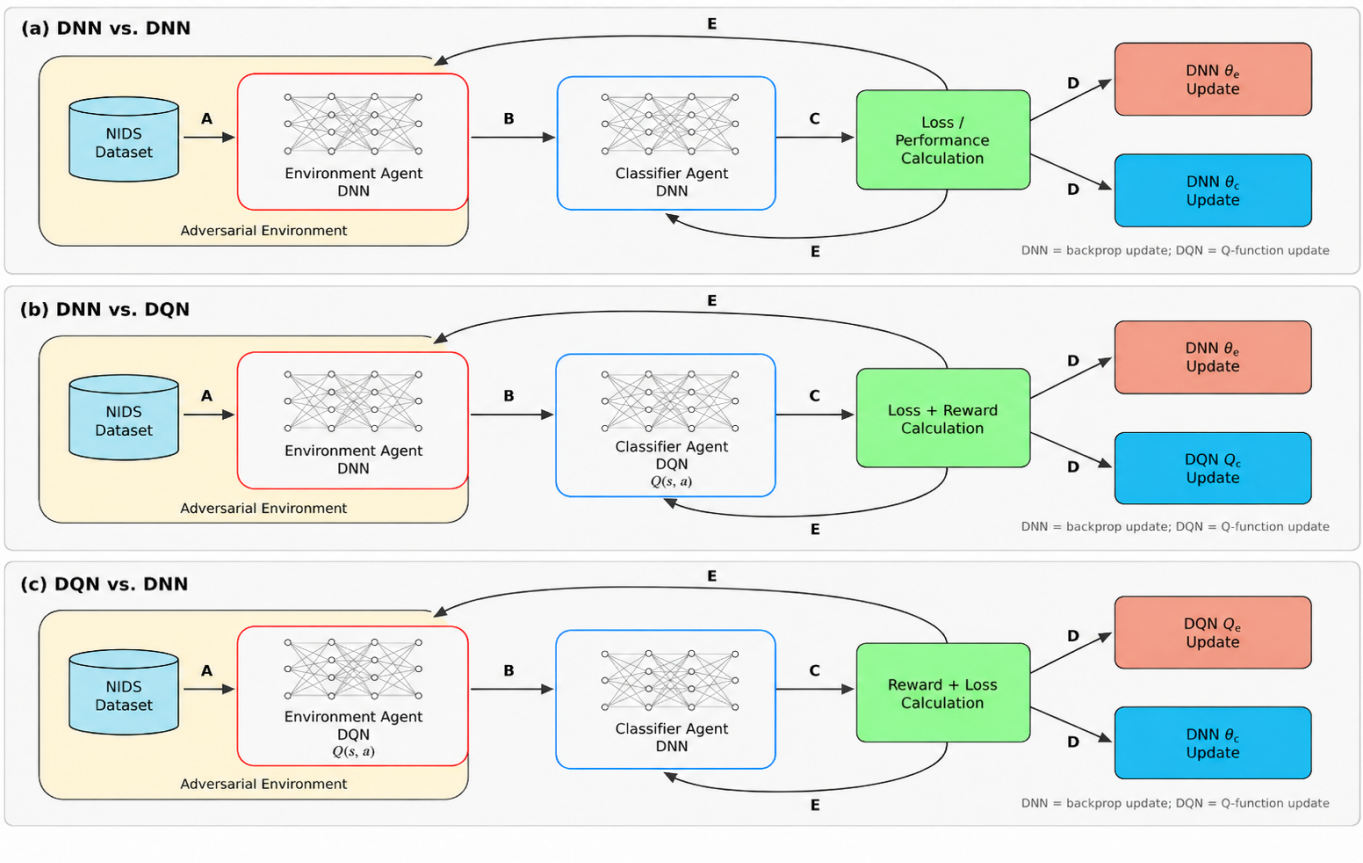


Fig. 3. Overview of the proposed adversarial-learning framework under three model pairings: (a) DNN versus DNN, (b) DNN versus DQN, and (c) DQN versus DNN. In each configuration, the adversarial agent samples network traffic from the networking dataset in step A and forwards the selected or modified traffic instance to the classifier agent in step B. The classifier agent then performs traffic classification or detection, and the resulting performance feedback is used to compute the relevant loss, reward, or combined feedback signal in step C. In step D, the feedback is routed to the appropriate update mechanism: DNN-based agents are updated through parameter optimization, while DQN-based agents are updated through their respective Q-function. The updated parameters or policies are then returned to the corresponding agents in step E, allowing the adversarial and classifier agent to continue learning across recurrent iterations. Figured adapted from [39].

7) Evaluation metrics and statistical testing: After training, we evaluate the defender on the held-out test fold and compute accuracy, precision, recall, F1-score, and G-Mean as defined in Section IV-A. Metrics are averaged across folds and reported with standard deviations. To assess whether observed differences between pairings are robust, we additionally apply paired statistical t-tests at an $\alpha = 0.05$ comparing the per-fold metric values of each method against alternatives, and we annotate statistically significant improvements in our result tables.

IV. Numerical Analysis

A. Performance Metrics

To evaluate the effectiveness of our approach, we report accuracy, precision, recall, F1-score, and geometric mean (G-Mean). Accuracy is computed as $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$, while the F1-score is defined as $\text{F1} = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$. Precision and recall are given by $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$, respectively, and G-Mean is calculated as $\text{G-Mean} = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{FP+TN}}$. For multiclass experiments, precision, recall, F1-score, and G-Mean are computed using macro averaging across the defender's coarse classes, while accuracy is computed globally across all observations. All experiments are executed on a workstation equipped with an Intel Core i7 CPU and 16 GB of RAM.

B. Traditional Local Area Network Scenario

TABLE I. Performance Comparison Across Defender/Attacker Pairings Using Cross-Validation for the Traditional LAN Scenario Using the CIC-IDS 2017 Dataset. The Best Results Per Metric are Bolded. Statistically Significant Improvements at $\alpha = 0.05$ (Paired T-Test Across Folds Versus the Other Two Models) are Denoted with an Asterisk (*)

Model	Accuracy	Recall	Precision	F1-Score	G-Mean
Both DNNs	0.9436 (0.0165)	0.4272 (0.0629)	0.4149 (0.0995)	0.4065 (0.0604)	0.6336 (0.0744)
DQN Defender	0.9668 (0.0037)	0.5287 (0.0105)	0.4915 (0.0601)	0.4865 (0.0211)	0.6961 (0.0425)
DNN Defender	**0.9828*** **(0.0090)**	**0.8168*** **(0.0762)**	**0.6539*** **(0.0696)**	**0.6775*** **(0.0682)**	**0.8048*** **(0.0456)**

Table I provides an overview of the performance metrics achieved when both defender and attacker utilize a DNN, when the defender uses a DQN, and attacker uses a DNN,

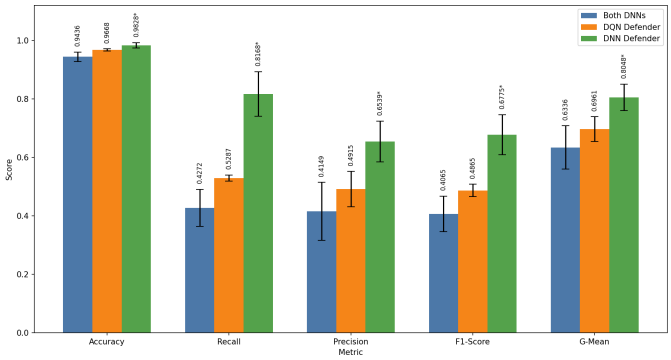


Fig. 4. Bar chart of the results of the model pairings in the LAN scenario when using the CIC-IDS 2017 dataset. Statistically significant results are indicated with an asterisk (*).

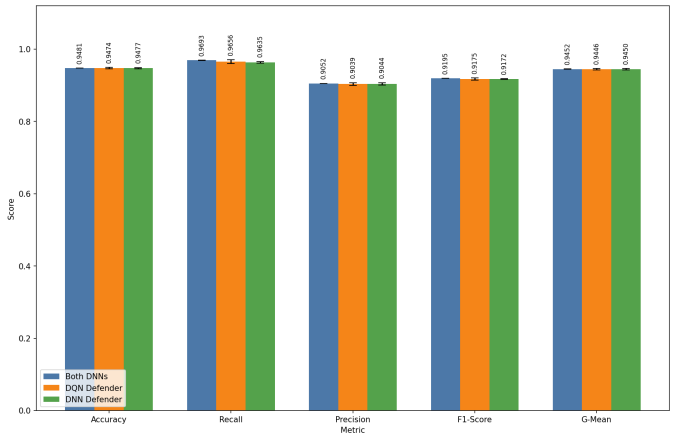


Fig. 5. Bar chart of the results of the model pairings in the ICS/SCADA scenario when using the DNP3 dataset. Statistically significant results are indicated with an asterisk (*).

and when the defender uses a DNN and the attacker uses a DQN for the traditional LAN scenario using the CIC-IDS 2017 dataset. Across all metrics in the CIC-IDS 2017 LAN scenario, the **DNN Defender** pairing achieved the strongest mean performance, statistically outperforming other models with respect to all of the performance metrics. Notably, we observe that the DNN defender statistically outperformed the other attacker-defender pairings by over 10% in metrics other than accuracy. The largest differences were seen in the recall class; this observation suggests having a DNN defender and DQN attacker yielded a lower false negative rate compared to the other models. Our results are depicted in Fig. 4.

C. ICS/SCADA Scenario

TABLE II. PERFORMANCE COMPARISON ACROSS DEFENDER/ATTACKER PAIRINGS USING CROSS-VALIDATION FOR THE ICS/SCADA SCENARIO USING THE DNP3 DATASET. THE BEST RESULTS PER METRIC ARE BOLDED. STATISTICALLY SIGNIFICANT IMPROVEMENTS AT $\alpha = 0.05$ (PAIRED T-TEST ACROSS FOLDS VERSUS THE OTHER TWO MODELS) ARE DENOTED WITH AN ASTERISK (*)

Model	Accuracy	Recall	Precision	F1-Score	G-Mean
Both DNNs	**0.9481 (0.0000)**	**0.9693 (0.0007)**	**0.9052 (0.0002)**	**0.9195 (0.0001)**	**0.9452 (0.0001)**
DQN Defender	0.9474 (0.0021)	0.9656 (0.0049)	0.9039 (0.0036)	0.9175 (0.0029)	0.9446 (0.0022)
DNN Defender	0.9477 (0.0017)	0.9635 (0.0024)	0.9044 (0.0031)	0.9172 (0.0011)	0.9450 (0.0020)

Table II provides an overview of the performance metrics achieved when both defender and attacker utilize a DNN, when the defender uses a DQN, and attacker uses a DNN, and when the defender uses a DNN and the attacker uses a DQN for the ICS/SCADA scenario using the DNP3 dataset. In the ICS/SCADA scenario using the DNP3 dataset, having **both DNNs** for the attacker and defender produced the best mean performance across all reported metrics. Importantly, all three configurations exhibit extremely low fold-to-fold variability; standard deviations ≤ 0.0049 across metrics, indicating that DNP3 is a comparatively stable dataset in which supervised decision boundaries already capture most of the separability available in the features. While having **both DNNs** provided the best performance; the performance was not statistically significant. Our results are depicted in Fig. 5.

D. IoT Network Scenario

TABLE III. PERFORMANCE COMPARISON ACROSS DEFENDER/ATTACKER PAIRINGS USING CROSS-VALIDATION FOR THE IOT ENVIRONMENT SCENARIO USING THE ACI-IOT DATASET. THE BEST RESULTS PER METRIC ARE BOLDED. STATISTICALLY SIGNIFICANT IMPROVEMENTS AT $\alpha = 0.05$ (PAIRED T-TEST ACROSS FOLDS VERSUS THE OTHER TWO MODELS) ARE DENOTED WITH AN ASTERISK (*)

Model	Accuracy	Recall	Precision	F1-Score	G-Mean
Both DNNs	0.9494 (0.0154)	0.5662 (0.0101)	0.5665 (0.0126)	0.5659 (0.0122)	0.7470 (0.0100)
DQN Defender	0.8360 (0.3185)	0.6298 (0.3211)	0.6621 (0.1605)	0.6067 (0.2958)	0.7928 (0.1326)
DNN Defender	**0.9906 (0.0066)**	**0.7898 (0.0061)**	**0.7896 (0.0100)**	**0.7896 (0.0080)**	**0.8872 (0.0065)**

Table III provides an overview of the performance metrics achieved when both defender and attacker utilize a DNN, when the defender uses a DQN, and attacker uses a DNN, and when the defender uses a DNN and the attacker uses a DQN for the IoT environment scenario using the ACI-IoT 2023 dataset. For the ACI-IoT network scenario, having a **DNN defender** with a DQN attacker achieved the strongest mean performance across all metrics with respect to traffic classification for the defender. The results of the **DNN defender** remain highly stable across folds, with the standard deviations ≤ 0.0100 for all metrics, suggesting more reliable generalization in this heterogeneous and imbalanced IoT setting. Results were not statistically significant, however. Our results are depicted in Fig. 6.

V. DISCUSSION

In the traditional LAN setting (CIC-IDS 2017), the DNN defender configuration achieved the best overall performance and produced statistically significant improvements across all reported metrics. The largest gains occurred for recall and F1-score, indicating that this pairing was most effective at detecting attacks rather than merely achieving high overall accuracy in an imbalanced setting. This suggests that, in LAN traffic where the feature space is rich but the attack classes can be sparse, a strong supervised decision boundary for the

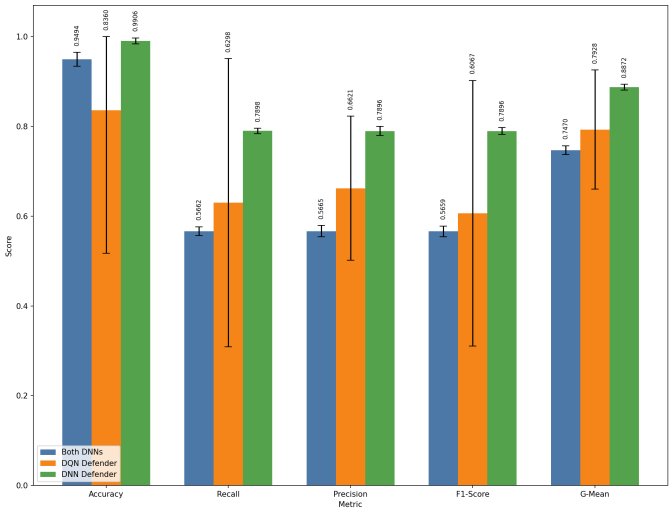


Fig. 6. Bar chart of the results of the model pairings in the IoT environment scenario when using the ACI-IoT dataset. Statistically significant results are indicated with an asterisk (*).

defender can remain robust even when the attacker adaptively influences the mixture of encountered samples.

In the ICS/SCADA setting (DNP3), having both defender and attacker utilize a normal DNN configuration produced the strongest and most stable results across every metric. Unlike the LAN and IoT datasets, DNP3 is substantially smaller and more controlled, and we used the balanced splits provided with the dataset. Under these conditions, supervised learning appears sufficient to capture the discriminative structure of the data, while reinforcement learning-based interaction provides limited additional benefit.

In the IoT setting (ACI-IoT), having a defender DNN in conjunction with an attacker DQN again achieved the best results and remained consistently stable across folds. In contrast, the reversal of the configuration in which the defender utilized a DQN and the attacker used a DNN displayed very high fold-to-fold variance. Our DQN defender failed to consistently learn an optimal policy for detecting network threats, resulting in substantial fold-to-fold variability.

A central finding across these experiments is therefore that reinforcement learning is not uniformly advantageous simply because it provides an adaptive learning mechanism. Instead, its effect depends on the role in which it is deployed and on the characteristics of the underlying network environment. In the CIC-IDS 2017 and ACI-IoT scenarios, the strongest mean performance was obtained when reinforcement learning was assigned to the attacker while the defender retained a supervised DNN, whereas the symmetric DNN configuration performed best in the more controlled DNP3 environment. These results suggest that the interaction among attacker capability, defender learning mechanism, and traffic heterogeneity can be as important as the selection of the classifier itself when evaluating intelligent NIDS.

We do acknowledge certain limitations with our work. The attacker controls the mixture of existing labeled traffic rather than generating new feature-space perturbations, and the evaluation does not include random, uniform, empirical-frequency, or hardest-class selection baselines. Consequently, some observed attacker advantage may reflect repeated selection of naturally difficult attack classes. Stratified flow-level cross-validation does not guarantee separation of records originating from the same sessions, hosts, or capture periods; the results therefore demonstrate flow-level rather than full temporal generalization. Despite these limitations, the proposed framework provides a useful basis for evaluating how asymmetric attacker–defender learning strategies influence intrusion-detection performance across heterogeneous network environments.

VI. Conclusion

Our work introduced an adversarial asymmetric learning framework for network intrusion detection that explicitly models an adaptive attacker alongside a defender, enabling controlled comparison of symmetric and asymmetric learning configurations. We formulated the interaction as a two-agent stochastic game in which the attacker influences the traffic mixture through fine-grained attack selection while the defender performs coarse-grained flow classification, and we evaluated DNN- and DQN-based agents under stratified 10-fold cross validation across three representative network environments: CIC-IDS 2017 (LAN), DNP3 (ICS/SCADA), and ACI-IoT (IoT).

Across experiments, we found that the best-performing pairing depends on the dataset regime. In the LAN and IoT settings, the DNN Defender configuration achieved the strongest detection performance and remained stable across folds, yielding substantial gains in sensitivity-focused metrics (recall, F1-score, and G-Mean). In the ICS/SCADA setting, the symmetric Both DNNs configuration produced the best and most consistent results, suggesting that supervised learning can be sufficient in smaller, more controlled and balanced environments. Conversely, reinforcement learning-based defenders exhibited elevated variance in heterogeneous and imbalanced settings, highlighting the importance of stability considerations when applying value-based deep reinforcement learning to IDS classification under adversarial pressure, as in the ACI-IoT dataset. We thus demonstrate that the effectiveness of adaptive learning in NIDS depends on both the role of the learning agent and the characteristics of the network environment.

Overall, the results provide two primary takeaways. First, explicitly modeling an adaptive attacker can materially change measured IDS performance relative to static evaluation, particularly for minority-class detection. Second, supervised defenders remain highly competitive under adversarial traffic-mixture control, while reinforcement learning-based defenders may require additional stabilization and richer reward models to realize consistent benefits. Several extensions can strengthen the realism and utility of this framework. A first direction is to incorporate cost-sensitive or asymmetric utilities such as including heavier penalties for false positives in high-volume IoT traffic or higher rewards for detecting high-impact ICS attacks. A second direction is to model temporality and partial observability more explicitly so that reinforcement learning can exploit genuine sequential structure. A third direction is to expand the adversary model beyond mixture control to include constrained feature-space perturbations or traffic

synthesis, bridging this framework with adversarial machine learning methods.

Declaration of Generative AI and AI-Assisted Technologies in the Manuscript Preparation Process

During the preparation of this work, the author utilized the generative AI service ChatGPT version 5.5 in order to improve the language and readability of the content. After using this service, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

[1] P. Sharikov, "Contemporary cybersecurity challenges," in *The Implications of Emerging Technologies in the Euro-Atlantic Space: Views from the Younger Generation Leaders Network*. Springer, 2023, pp. 143–157.

[2] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Wiley, 2020.

[3] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (ids)," *International journal of information security*, vol. 22, no. 5, pp. 1125–1162, 2023.

[4] H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, and H. Wang, "Large language models for cyber security: A systematic literature review," *ACM Trans. Softw. Eng. Methodol.*, Sep. 2025. [Online]. Available: https://doi.org/10.1145/3769676

[5] H. Alqahtani and G. Kumar, "A comprehensive review of generative AI techniques and their impact on cybersecurity," *Soft Computing*, vol. 29, no. 13–14, pp. 4945–4982, 2025.

[6] A. M. K. Adawadkar and N. Kulkarni, "Cyber-security and reinforcement learning—a brief survey," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105116, 2022.

[7] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[8] V. Kelli, P. Radoglou-Grammatikis, A. Sesis, T. Lagkas, E. Fountoukidis, E. Kafetzakis, I. Giannoulakis, and P. Sarigiannidis, "Attacking and defending dnp3 ics/scada systems," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2022, pp. 183–190.

[9] E. A. Nack, M. C. McKenzie, and N. D. Bastian, "Aci-iot-2023: A robust dataset for internet of things network security analysis," in *MILCOM 2024-2024 IEEE Military Communications Conference (MILCOM)*. IEEE, 2024, pp. 1–6.

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[12] L. Li, Y. Lu, G. Yang, and X. Yan, "End-to-end network intrusion detection based on contrastive learning," *Sensors*, vol. 24, no. 7, p. 2122, 2024.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.

[14] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[15] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104 650–104 675, 2020.

[16] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[17] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, no. 3, p. 41, 2022.

[18] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports*, vol. 12, p. 15370, 2022.

[19] M. A. Hossain, "Deep Q-learning intrusion detection system (DQ-IDS): A novel reinforcement learning approach for adaptive and self-learning cybersecurity," *ICT Express*, vol. 11, no. 5, pp. 875–880, 2025.

[20] R. Kanimozhi and P. S. Ramesh, "Deep reinforcement learning-based intrusion detection scheme for software-defined networking," *Scientific Reports*, vol. 15, p. 38827, 2025.

[21] C. Rookard, "Spider: Space satellite intrusion detection using explainable reinforcement learning for next-generation satellite communication networks," in *Telecom*, vol. 7, no. 1. MDPI, 2026, p. 3.

[22] J. F. Cevallos Moreno, A. Rizzardi, S. Sicari, and A. Coen-Porisini, "Deep reinforcement learning for intrusion detection in internet of things: Best practices, lessons learnt, and open challenges," *Computer Networks*, vol. 236, p. 110016, 2023.

[23] A. Gueriani, H. Kheddar, and A. C. Mazari, "Deep reinforcement learning for intrusion detection in IoT: A survey," arXiv preprint, 2024.

[24] W. Yang, A. Acuto, Y. Zhou, and D. Wojtczak, "A survey for deep reinforcement learning based network intrusion detection," arXiv preprint, 2024.

[25] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, vol. 159, pp. 96–109, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618311216

[26] K. Hammar and R. Stadler, "Finding effective security strategies through reinforcement learning and self-play," arXiv preprint, 2020.

[27] R. Elderman, L. J. J. Pater, A. S. Thie, M. M. Drugan, and M. M. Wiering, "Adversarial reinforcement learning in a cyber security simulation," in *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, INSTICC. SciTePress, 2017, pp. 559–566.

[28] K. He, D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[29] H. A. Alatwi and C. Morisset, "Adversarial machine learning in network intrusion detection domain: A systematic review," arXiv preprint, 2021.

[30] J. Vitorino, I. Praça, and E. Maia, "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection," *Computers & Security*, vol. 134, p. 103433, 2023.

[31] R. Sheatsley, N. Papernot, M. J. Weisman, G. Verma, and P. McDaniel, "Adversarial examples for network intrusion detection systems," *Journal of Computer Security*, vol. 30, no. 5, pp. 727–752, 2022.

[32] B.-E. Zolbayar, R. Sheatsley, P. McDaniel, M. J. Weisman, S. Zhu, S. Zhu, and S. Krishnamurthy, "Generating practical adversarial network traffic flows using NIDSGAN," arXiv preprint, 2022.

[33] I. Debicha, B. Cochez, T. Kenaza, T. Debatty, J.-M. Dricot, and W. Mees, "Adv-bot: Realistic adversarial botnet attacks against network intrusion detection systems," *Computers & Security*, vol. 129, p. 103176, 2023.

[34] C. Rookard, "Evaluating adversarial resilience of deep reinforcement learning algorithms for network intrusion detection," in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, 2026, pp. 1–6.

[35] A. McCarthy, E. Ghadafi, P. Andriotis, and P. Legg, "Defending against adversarial machine learning attacks using hierarchical learning: A case study on network traffic attack classification," *Journal of Information Security and Applications*, vol. 72, p. 103398, 2023.

[36] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. Wiley, 1994.

[37] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[39] C. Rookard and A. Khojandi, "Rriot: Recurrent reinforcement learning for cyber threat detection on iot devices," *Computers & Security*, vol. 140, p. 103786, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824000877

Assisted Robot Arm with Voice Recognition for Digital Healthcare

Ting Zhang[1]*, Oanh Phan[2], Jiang Lu[3],
Ting Zhang-Program of Computer Science, University of Houston Downtown, Houston, USA[1]
Oanh Phan-Program of Computer Science, University of Houston Downtown, Houston, USA[2]
Jiang Lu-Program of Computer Engineering, University of Houston Downtown, Houston, USA[3]

***Abstract*—Healthcare systems face increasing pressure from nursing shortages, growing patient-care demands, and the substantial amount of time healthcare professionals spend performing routine and repetitive bedside activities. These challenges can increase caregiver workload and reduce the time available for clinical assessment, decision-making, and direct patient interaction. Patients with limited mobility may also depend on caregivers for relatively simple activities, such as retrieving objects, obtaining drinking water, or interacting with their immediate environment, which can lead to delayed assistance when healthcare personnel are occupied with higher-priority responsibilities. Motivated by these challenges, this study investigates the use of collaborative robotics and intelligent natural-language interaction to provide supplementary assistance for predictable and repetitive patient-care tasks. An intelligent robotic patient-assistance framework is developed by integrating collaborative robotic manipulation, voice-based natural-language interaction, and non-contact physiological monitoring. The system employs a UFACTORY xArm7 collaborative manipulator to perform representative bedside assistance tasks, including cleaning a designated area, pouring water, delivering medicine, and measuring body temperature. A Python-based software architecture coordinates robot control, task execution and user interaction. To provide hands-free and intuitive control, Faster-Whisper is used for continuous speech-to-text processing, while a Large Language Model (LLM) interprets the transcribed commands and maps different natural-language expressions to predefined robotic tasks.**

***Keywords*—Healthcare; Robot Arm; LLMs; automation; voice recognition; healthcare innovation**

I. Introduction

Healthcare is an ever-evolving field that continuously pursues innovative approaches to improve patient outcomes and enhance the efficiency of healthcare operations. Rapid technological advancements have created new opportunities to transform healthcare delivery, with robotics emerging as a promising solution for supporting clinical tasks, improving workflow efficiency, and advancing the overall quality of patient care.

A. Challenges of Traditional Healthcare and Patient Care

Traditional healthcare systems depend heavily on human healthcare professionals to perform both clinical and routine patient-care activities, creating significant challenges as patient populations grow and workforce shortages persist. The global shortage and uneven distribution of nursing personnel can increase individual workloads and limit the time available for direct patient interaction and continuous monitoring [1]. High workloads and repetitive responsibilities can contribute to physical and psychological fatigue among healthcare workers. Nurse burnout has also been associated with missed care, medication errors, patient safety incidents, and reduced patient satisfaction [2]. Furthermore, inadequate staffing and limited resources may result in delayed or missed nursing interventions [3]. Without robotic manipulators, many repetitive physical tasks must be performed manually, limiting opportunities to automate routine bedside assistance and potentially reducing the time clinicians can devote to tasks requiring professional judgment and interpersonal interaction. At the same time, traditional healthcare information systems generally provide limited natural-language interaction; without large language models (LLMs), which is a new, innovative and powerful technique created from recent years. Recent studies indicate that LLMs have potential applications in patient communication, clinical documentation, and information processing [4], [5]. Therefore, combining large language models with robotic manipulation offers a potential approach for reducing repetitive workload while enabling more intuitive interaction between patients, healthcare personnel, and assistive robotic systems.

1) Nursing shortages and high patient-to-nurse ratios: Nursing shortages and excessive nursing workloads can contribute to fatigue, emotional exhaustion, reduced job satisfaction, and intentions to leave the profession. A systematic review and meta-analysis found that nurse burnout was associated with poorer patient-safety outcomes, lower perceived care quality, and reduced patient satisfaction [6]. Workforce shortages and burnout may therefore form a reinforcing cycle: insufficient staffing increases workload, elevated workload contributes to burnout, and burnout increases turnover, thereby placing further pressure on the remaining workforce [7], [8].

2) Economic burden of nursing services: An adequately staffed nursing workforce represents a substantial but necessary component of healthcare expenditure. The institutional cost of employing a nurse extends beyond base salary and includes health insurance, retirement contributions, paid leave, continuing education, recruitment, orientation, and overtime compensation. Because hospitals and residential-care facilities operate continuously, multiple employees are required to provide uninterrupted coverage for a single position across various shifts [9], [10].

3) Delayed and missed patient care: For patients with impaired mobility or upper-limb function, delayed assistance can have consequences beyond temporary inconvenience. A

*Corresponding author.

patient may remain unable to drink, communicate, retrieve an essential object, or adjust the immediate environment until a caregiver becomes available [11], [12]. Some patients may attempt these activities without assistance, potentially increasing the risk of falls, overextension, equipment displacement, or other preventable injuriecite. Technologies that respond to structured, low-risk requests could therefore improve patient independence while reducing interruptions to nurses performing higher-priority clinical work [14], [13].

4) Limitations of exclusively human-dependent care: Direct human interaction remains an essential component of nursing because it enables clinical judgment, emotional support, and individualized patient care. However, relying exclusively on nurses to perform both complex clinical procedures and routine physical tasks presents several operational challenges. Frequent bedside contact increases occupational exposure to infectious diseases despite established infection-control measures [15], [16]. Repetitive activities such as retrieving supplies, repositioning equipment, and assisting with routine patient requests also contribute to physical fatigue and work-related musculoskeletal disorders, which remain common among nursing professionals [18], [17]. Furthermore, the availability of nursing staff is constrained by workload, patient acuity, and emergency situations, making immediate responses to routine requests difficult during busy clinical periods. Consequently, assigning repetitive logistical tasks solely to nurses may reduce overall workflow efficiency and limit the time available for assessment, communication, and direct clinical care. These limitations highlight the potential value of assistive robotic systems that can undertake predictable physical tasks while allowing nurses to focus on responsibilities requiring professional expertise and interpersonal interaction [19].

B. Literature Review of Robotic Assisted Patient Care and Research About LLMs

1) Existed technologies for robotic assisted patient care: Healthcare robotics has gradually evolved from autonomous transportation platforms to intelligent assistive systems capable of interacting directly with patients [20]. Early hospital robots were primarily developed for logistics, including the transportation of medication, laboratory samples, and medical supplies, with the objective of reducing caregiver travel and improving operational efficiency. Advances in autonomous navigation, perception, and human–robot interaction have subsequently enabled robots to perform bedside assistance, allowing them to operate safely in clinical environments while collaborating with healthcare professionals [21], [22].

Recent research has increasingly focused on robotic manipulators because physical interaction is required for many patient-care activities. Compared with mobile service robots, robotic arms provide dexterous manipulation that enables direct assistance during activities of daily living [23], [24]. Their applications include object retrieval, assistive feeding, and manipulation of bedside devices for individuals with limited upper-limb function. Shared-autonomy approaches have further improved usability by combining autonomous robot motion with user intention, thereby reducing operator workload while maintaining safe interaction during manipulation tasks [25].

2) Human–robot interaction and user acceptance: The successful deployment of assistive robotic systems in healthcare depends not only on technical performance but also on their acceptance by patients and healthcare professionals. Studies have shown that user acceptance is strongly influenced by perceived safety, trust, ease of interaction, and the robot's ability to complement existing clinical workflows rather than replace human caregivers in STEM researches [26]. Although robotic assistants can reduce repetitive physical workload and improve the availability of routine patient assistance, concerns remain regarding privacy, patient dignity, responsibility for system failures, and excessive reliance on automation. These issues highlight the importance of maintaining human supervision during robot operation and ensuring that clinical staff retain full authority over safety-critical decisions [27].

3) The innovative and powerful technology of LLMs: Large Language Models (LLMs) are increasingly being explored in healthcare to improve communication and support clinical workflows. They can assist with medical question answering, patient education, and the interpretation of clinical information, allowing healthcare professionals to access relevant knowledge more efficiently [28], [29]. LLMs can also support clinical documentation by generating or summarizing medical text from patient encounters, potentially reducing the administrative workload placed on healthcare providers. In patient-care environments, natural-language interaction may allow patients to express their needs conversationally rather than through predefined commands. When combined with assistive technologies such as robotic systems, this capability can help translate patient requests into structured instructions while maintaining human supervision for safety-critical decisions [30], [31].

4) LLMs' applications in healthcare about voice recognition: Large Language Models (LLMs) have emerged as an important artificial intelligence technology capable of understanding natural language and generating context-aware responses. Their ability to interpret complex instructions has supported their growing use in healthcare [28]. LLMs can provide a natural-language interface between users and intelligent healthcare systems, making communication more intuitive. When integrated with robotic systems, an LLM can interpret patient requests, such as "bring me some water," and translate their meaning into structured commands for subsequent robotic execution. This capability may support the development of healthcare assistants that respond more naturally to patient needs, although appropriate validation and human supervision remain necessary for clinical applications [30].

C. Objectives and Scope of the Project

This project introduces the concept of harnessing a collaborative UFACTORY xArm7 as an automated assistant in healthcare. This approach capitalizes on the robot's potential to enhance patient care and efficiency through its role in aiding patient care tasks and conducting vital sign monitoring. We utilized LLMs with the robot arm, to achieve voice recognition for delivery of patient needs. Leveraging the Python programming language for implementation, this project not only underscores the practical fusion of technology with healthcare but also explores the potential paradigm shift in patient care dynamics.

The first objective of this research is to design and implement a robotic patient-assistance framework using the UFACTORY xArm7 collaborative manipulator. The proposed framework enables the robotic arm to perform routine bedside assistance tasks through an integrated architecture that combines robotic manipulation, sensing, voice-based user interaction, and intelligent task execution. The system focuses on tasks such as pouring water, cleaning a designated area, delivering medicine, and taking temperature measurements. The overall objective is to demonstrate how collaborative robotic technology can provide reliable physical assistance while operating safely in environments shared with patients and healthcare professionals.

The second objective is to develop a voice-controlled human–robot interaction system that enables users to request assistance through natural spoken commands. A speech-to-text module is incorporated to capture the user's voice and convert spoken requests into textual information for further processing. This approach enables patients to interact with the robotic system without requiring physical access to a computer, touchscreen, or dedicated control interface, providing a more accessible method of interaction for users with limited mobility.

The third objective is to integrate a Large Language Model (LLM) into the voice-control framework for intelligent interpretation of user requests. The LLM is used to analyze recognized speech, identify the user's intended task, and map different natural-language expressions to predefined robotic actions. For example, expressions such as "Bring me some water," "Can I have some water?" and "I am thirsty" can be associated with the same water-assistance task. The LLM-based approach enables more flexible and intuitive communication by allowing users to express requests naturally rather than relying on a fixed set of predefined voice commands.

The fourth objective is to investigate autonomous object manipulation for routine patient-care applications. This work focuses on developing task-specific motion planning, grasping, and manipulation strategies that enable the xArm7 to retrieve, transport, position, and manipulate objects required for bedside assistance. The robotic system is intended to autonomously execute tasks identified through the voice-control framework, including water pouring, area cleaning, and medicine delivery. By automating predictable and repetitive manipulation activities, the proposed framework aims to reduce the need for direct human intervention while supporting routine patient assistance.

The fifth objective is to integrate non-contact physiological monitoring into the robotic assistance platform. In addition to performing physical manipulation tasks, the system is intended to support non-contact sensing technologies for acquiring physiological information, with particular emphasis on body temperature measurement. Integrating physiological monitoring with robotic manipulation allows the same platform to provide both physical assistance and basic patient observation. This combined capability establishes a foundation for future expansion toward additional non-contact physiological measurements and more comprehensive intelligent patient-monitoring systems.

II. Materials and Methods

A. Hardware and Software Environment

The experimental platform employs a UFACTORY xArm7 collaborative robotic manipulator as the primary hardware for autonomous patient-assistance and vital-sign monitoring tasks. The xArm7 is a seven-degree-of-freedom (7-DoF) collaborative robot designed to provide kinematic redundancy, enabling flexible motion planning and obstacle avoidance in constrained environments while maintaining high positioning accuracy [32]. The robot offers a maximum payload capacity of 3.5 kg, a reach of 700 mm, and a repeatability of ± 0.1 mm, making it suitable for manipulation tasks that require precise positioning and safe human–robot interaction. Compared with conventional six-axis manipulators, the additional degree of freedom allows the robot to maintain multiple feasible arm configurations for the same end-effector pose, thereby improving dexterity and reducing the likelihood of singular configurations during operation [33].

The xArm7 is controlled through the UFACTORY controller using the official Python Software Development Kit (SDK), which provides high-level interfaces for joint-space motion, Cartesian trajectory planning, inverse kinematics, digital input/output (I/O), and gripper control. Communication between the host computer and the robot controller is established through a TCP/IP Ethernet connection, enabling seamless integration with external software developed in Python and Robot Operating System (ROS/ROS 2) environments. During system deployment, safety mechanisms including collision detection, self-collision avoidance, configurable safety boundaries, and emergency stop functions are enabled to ensure safe operation when the robot performs collaborative tasks in close proximity to users [34]. These hardware and software capabilities provide a flexible experimental platform for implementing intelligent robotic assistance systems in healthcare applications.

The Robot application was executed on an MSI gaming computer equipped with an Intel Core i7-13700F processor, an NVIDIA GeForce RTX 4070 graphics card with 12 GB of dedicated video memory, 32 GB DDR5 RAM, and a 1 TB NVMe solid-state drive running Windows 11 (64-bit) [35], [36], [37], [38]. These specifications exceed the recommended hardware requirements for real-time VR rendering, ensuring smooth operation and stable frame rates during the experimental evaluations.

We created a voice control component based on LLMs. The voice command was the input of a speech to text module, and LLMs created the intent classification, that directs the root to perform certain tasks. The speech to text component was through python text to speech package and the LLMs was GPT package.

We created python algorithms to precisely control the robot arm's movements and interactions, forming the programming and control stage. During the programming and control stage, sophisticated control algorithms are developed using the Python programming language to enable precise and reliable operation of the robotic system. Python serves as the primary software development platform because of its extensive ecosystem of robotics libraries, numerical computation packages, and compatibility with the UFACTORY Software Development Kit (SDK). The developed software

is responsible for generating motion commands, performing inverse kinematics calculations, planning collision-free trajectories, and coordinating end-effector operations throughout each experimental task. Communication between the host computer and the robot controller is established through a TCP/IP Ethernet connection, allowing real-time transmission of motion commands and continuous monitoring of the robot's operating status.

B. System Design and Experimental Steps

The design goals are to create a robotic assistance framework, that is capable of Autonomous object manipulation with non-contact physiological monitoring. The software is capable of recognizing user input through voice recognition, mapping the task from the voice control, performing guided tasks, gathering data and recording of robot movement videos. Fig. 1 shows a diagram of the design objectives. The system design involves the following components. The Robotic Arm is located on a mobile table for easy manipulation, the connected computer serves as a hub for robot control and data aggregation. Python based tasks such as cleaning up, pouring water, delivering of medicine and vital sign measurement were established. On the user interface, the user can select tasks and record the robot actions during the tasks.

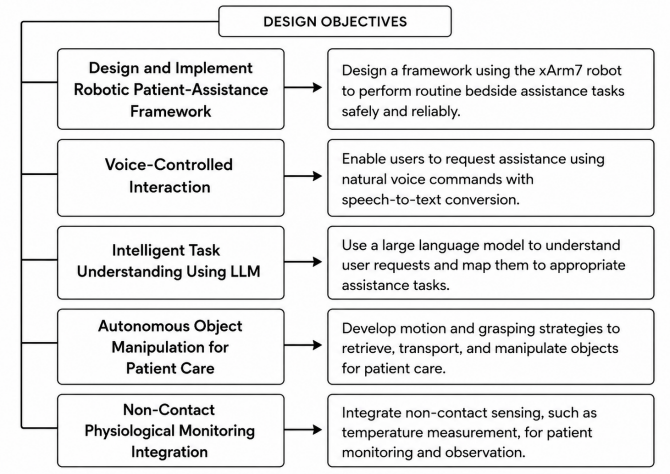


Fig. 1. Design objectives.

The software focused control package consisted the voice recognition component, the task identification component, and the robot task direction component. The voice recognition component utilized python text to speech package. The task identification component involves GPT package for intent mapping. The user interface built connections to direct robot for particular tasks.

The hardware related project methodology involves a comprehensive approach that comprises several key stages and assessment procedures. First, the UFACTORY Robot Arm is equipped with specialized end-effectors that can be designed for healthcare applications, and this involves the initial design and integration phase. Next, algorithms are developed using the Python programming language to precisely control the robot arm's movements and interactions, forming the programming and control stage. To ensure reliability and safety, virtual environments are painstakingly created for thorough testing of the robot's actions before real-world deployment in the simulation and rigorous testing phase. Additionally, a user-friendly interface is carefully designed to provide healthcare professionals with seamless control over the robot arm's activities, constituting the user interface design phase. Throughout the process, data is systematically collected in both simulated and real healthcare scenarios, enabling in-depth analysis during the data acquisition and analysis stage. Finally, the performance of the automated assistant is evaluated using various metrics, including accuracy, operational efficiency, and we analyzed its impact on patient care, as part of the evaluation and impact assessment phase. Fig. 2 shows the system diagram involving both hardware and software.

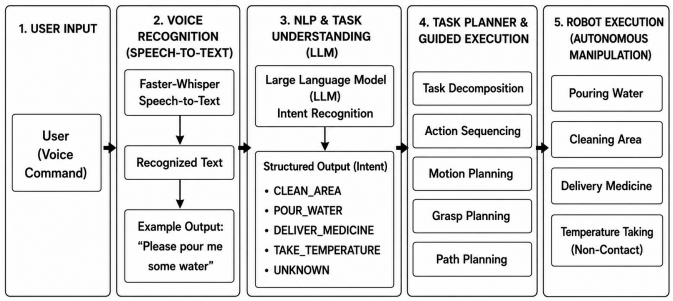


Fig. 2. System design diagram.

C. LLM-Based Natural Language Control Software Design

The speech-to-text component was designed to provide continuous voice recognition for natural human–robot interaction. Faster-Whisper was selected as the primary speech recognition framework because it provides a suitable balance between transcription accuracy and computational efficiency, making it appropriate for near-real-time robotic applications. The framework can be integrated into the Python-based control architecture to capture audio signals from a microphone, process the recorded speech, and convert the resulting audio into textual information. During operation, microphone input is continuously acquired and processed so that recognized speech can be displayed in real time and subsequently forwarded to the natural-language processing module. This continuous recognition approach allows users to provide successive verbal commands without manually restarting the speech-recognition process for each interaction. The transcribed text serves as the interface between the speech-recognition and Large Language Model components, where the recognized user request is further analyzed to determine the corresponding robotic task. If speech cannot be reliably recognized, the system can reject the input or request another command rather than initiating an uncertain robotic action. This implementation establishes a continuous voice-processing pipeline consisting of audio acquisition, speech recognition, text generation, intent interpretation, and robotic task selection, thereby supporting intuitive hands-free interaction with the patient-assistance system.

The LLM-based natural language control software is designed to provide an intuitive communication interface between the user and the robotic patient-assistance system.

Spoken commands are first captured through a microphone and converted into text using a speech-to-text module such as Faster-Whisper. The recognized text is then processed by a Large Language Model (LLM) to interpret the user's intent and associate different natural-language expressions with predefined robotic tasks. For example, commands such as "Bring me some water," "Can I have some water?" or "I am thirsty" can be interpreted as a water-assistance request. The LLM produces a structured task identifier corresponding to supported functions such as water pouring, area cleaning, medicine delivery, or temperature measurement. This task identifier is subsequently passed to the robot-control software, where predefined and validated motion routines are executed by the xArm7. By separating natural-language interpretation from low-level robotic control, the proposed software architecture provides flexible voice interaction while maintaining predictable and controlled robot behavior. Fig. 3 shows the diagram of voice recognition and LLM based task mapping code.

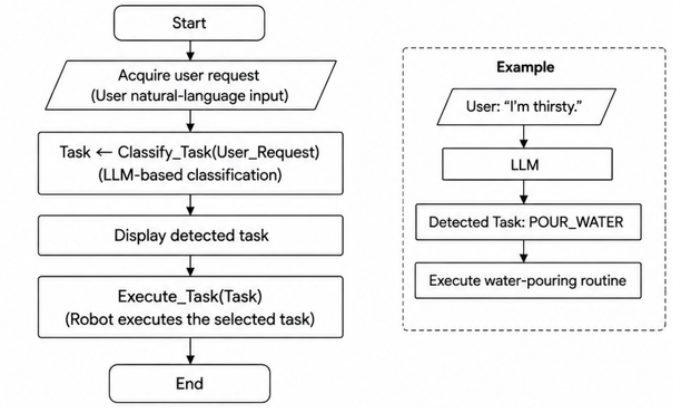


Fig. 3. Diagram of the process for the voice recognition and LLM-based task mapping.

D. Robot Installation and Initial Configuration

The installation of the UFACTORY xArm7 collaborative robotic system consists of several sequential steps to ensure safe and reliable operation. First, an appropriate workspace is defined to provide sufficient clearance for the robot's operating range and to prevent potential collisions with surrounding objects. The robotic arm is then securely mounted onto a stable work surface to ensure mechanical stability during operation. Next, the robotic arm is connected to the UFACTORY control box through the dedicated communication interface, followed by the connection of the control box to the power supply and the host computer using the required communication and power cables. After the hardware installation is completed, the system is powered on and initialized to verify proper communication between the robot controller and the host computer. Fig. 4 shows the connection of the robot arm with the computer that has the user interface, the robot control software component and robot task recording software component.

Following hardware installation, the robot undergoes an initial configuration and familiarization process using the *xArm Studio* software developed by UFACTORY. This software provides an integrated development environment for robot operation, programming, and system configuration. Users become

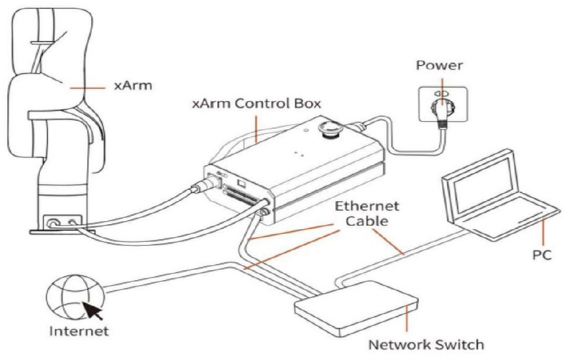


Fig. 4. Robot connection diagram.

familiar with the robot's hardware, control interface, operating modes, and workspace while verifying the robot's motion capability and communication status. Fig. 5 shows the central control unit of the system. Fig. 6 shows the XARM design interface. The software consists of five primary functional modules: *Live Control*, which enables manual manipulation of the robotic arm; *Blockly*, which supports graphical block-based programming; the *Python IDE*, which provides script-based robot programming and debugging; *Recording*, which allows robot trajectories to be recorded and replayed; and *Settings*, which manages system configuration, communication parameters, safety settings, and robot calibration. These integrated software tools facilitate rapid system deployment, robot programming, and experimental validation while providing a user-friendly interface for both novice and experienced users.



Fig. 5. Assisted robot Arm system - the central control unit.

The integration of the xArm presents several challenges, particularly in the realm of coordination. The mismatch between the coordinate systems of the xArm and the gripper introduces the potential for coordination issues during object manipulation. This misalignment can lead to inaccuracies and may result in failed attempts at grasping and pouring tasks. Additionally, the calibration process, essential for ensuring precise alignment between the two coordinate systems, proves to be a complex and time-consuming endeavor. Achieving perfect alignment demands meticulous adjustments, contributing to the overall intricacy of the setup.

Furthermore, an encountered challenge relates to IP configuration, where issues arose during power cycling. The IP



Fig. 6. The XARM studio design interface.

addresses of the computer and the control box were found to be on different network segments, posing a major hurdle. To address this, it became necessary to modify the computer's IP within the range of 192.168.1.1 - 192.168.1.255, with the tail number falling between 1 and 255, and ensuring it does not match the IP address of the control box. This adjustment was crucial for establishing a seamless network connection and overcoming the encountered IP configuration challenge.

E. Robot Control Software Design and Integration

We developed Python scripts that empower the UFACTORY robot arm to execute patient care tasks, ensuring precise and fluid movements. We created a software interface that bridges the gap between healthcare staff and the robot arm. This interface is designed to enable intuitive control, communication, and coordination, aligning seamlessly with the workflow of medical professionals. Additionally, the performance of the robot arm is recorded, with the goal for validation of calibration, and task completion testing.

Live Control offers the capability to manipulate the xArm's position and adjust its posture using XYZ coordinates within the base coordinate system for the arm and the tool coordinate system for the gripper. Fig. 7 shows the interface of the Live Control of the Assisted Robot Arm.

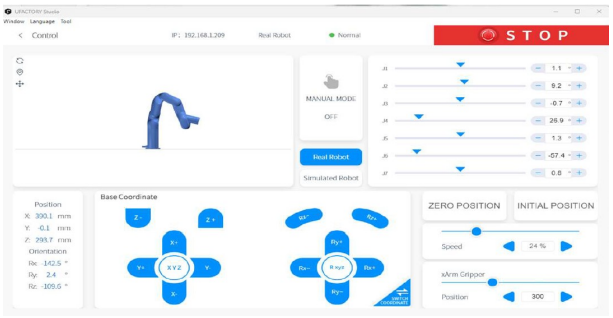


Fig. 7. Interface of live control of the assisted robot Arm.

Blockly, a graphical programming tool, simplifies programming tasks by allowing users to drag and drop code blocks to control the robotic arm, input, and output. Fig. 8 shows the Blockly interface for experimenting kinematics parameters. The diagram illustrates the block-based programming workflow used to control the UFACTORY xArm7 robotic joints and associated end-effector functions. The process begins with the Move Servo (Joint Motion) command, which specifies the target joint, desired angular position, starting angle, motion duration, speed, and execution mode. These parameters determine how an individual robot joint moves from its initial configuration to a defined target position. The workflow then incorporates the Set Analog Output command, which can be used to regulate auxiliary devices such as the gripper by assigning an output pin and control value. Subsequent joint movements can be generated by repeating the servo command with updated target parameters, allowing a sequence of coordinated robot poses to be constructed.

The final stage uses the Set End I/O Pin (Digital Output) command to control binary end-effector functions, such as opening or closing the gripper. The supporting reference tables summarize the principal command parameters and identify the seven joints of the xArm7, from the base joint to the distal wrist joints. Together, these programming elements provide a structured approach for defining joint trajectories, controlling end-effector operations, and coordinating sequential robot actions. The block-based interface therefore enables robot motion to be programmed without requiring extensive low-level coding, while still providing control over position, speed, timing, and I/O states.



Fig. 8. Blockly interface for experimenting kinematics parameters.

The Python IDE facilitates direct utilization of the Python SDK API for creating programs that instruct the robotic arm to perform particular tasks. after experimenting with blockly for key parameters, we program each task as a python program. Fig. 9 shows an example Python pseudocode for the task of pouring water.

III. Results

A. Results of the Natural Language Based LLM Enabled Task Mapping

Fig. 10 shows an example display of python code execution for the natural language based LLM enabled task mapping. The output demonstrates the successful operation of the LLM based natural language task classification module. The system receives transcribed user commands and interprets their semantic meaning to determine the corresponding predefined robotic task. For example, the statement "I'm thirsty" is classified as `POUR_WATER`, demonstrating that the system can infer the user's intention even when the requested action is not explicitly stated. Similarly, "Please clean the table" is mapped to `CLEAN_AREA`, while "Check my temperature" is classified as `TAKE_TEMPERATURE`. These results demonstrate that the LLM can associate different natural-language expressions with

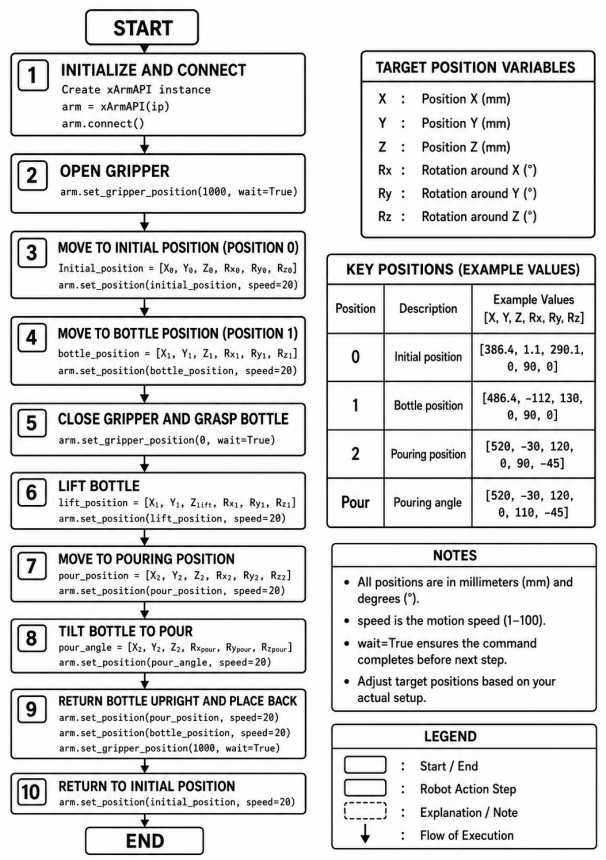


Fig. 9. Pseudo code and illustrations of the assisted robot Arm for pouring water.

standardized task identifiers, which can subsequently be passed to the robot-control module to initiate the appropriate predefined task routine.

```
You said: I'm thirsty
Detected task: POUR_WATER

You said: Please clean the table
Detected task: CLEAN_AREA

You said: Check my temperature
Detected task: TAKE_TEMPERATURE
```

Fig. 10. Python output for voice recognition and LLM based task mapping.

B. Assisted Robotic Arm Performance for Example Tasks and Use Cases

Empirical evidence forms the cornerstone of this research's contributions. A comprehensive account of the data was collected during task execution, including completion times, accuracy rates, and any observed safety considerations, was provided. These results were contextualized within the research's objectives and contribute to the overall evaluation of the robot arm's efficacy and feasibility in healthcare scenarios. The different tasks were divided into cleaning area, pouring Water, delivering medicine, taking body temperature.

1) Remove medical contaminated materials: The deployment of the UFACTORY xArm7 robotic arm and gripper for the removal of medical contaminated materials in healthcare settings had made significant strides, leveraging advanced robotic technology to enhance efficiency and safety. The deployment process began with the installation of the xArm Python Library, facilitating seamless communication with the robotic arm. Following this, the necessary modules were imported to enable comprehensive control and programming capabilities. An instance of the xArm API class was created, specifying the IP address or port of the xArm7 to establish a connection. Through the "connect" method, the xArm seamlessly linked to the designated system. The deployment strategy incorporated a loop to repeat the material removal task multiple times, allowing for systematic and repetitive operations. Positions of the xArm were dynamically adjusted within the loop to navigate the robot precisely to the targeted contaminated materials. Upon completing the task iterations, the xArm disconnected from the system, marking the successful execution of the deployment plan. This progress highlighted the successful integration of advanced robotic solutions for the efficient and controlled removal of medical contaminated materials in healthcare settings.

2) Deliver water / pouring water into the cup: In a healthcare context, implementing the precise and controlled pouring of water from a bottle into a cup using the UFACTORY xArm7 robotic arm equipped with a gripper offered a range of potential benefits. This application was particularly useful in scenarios where patients have difficulty performing such tasks independently due to physical limitations or health conditions. The robotic arm's precision allowed for accurate positioning over the cup, ensuring a controlled and spill-free pouring process. The gripper's dexterity enabled the arm to securely hold and manipulate the water bottle, providing a level of assistance that could contribute to patients' autonomy and well-being. By automating simple yet essential tasks like pouring water, the UFACTORY xArm7 and gripper combination enhanced the efficiency of healthcare facilities, allowing healthcare professionals to focus on more complex aspects of patient care while simultaneously promoting patients' independence and comfort. Developing a Python code implementation for pouring water from a bottle into a cup using the UFACTORY xArm7 robotic arm and gripper in a healthcare setting involved integrating precision and control to assist patients with limited mobility. The code initiated the robotic arm's movement to grasp the water bottle securely with the gripper, navigated to a predetermined position over the cup, and gradually tilted the bottle to pour water accurately. Implementing safety measures was crucial, ensuring a controlled pouring process that minimizes the risk of spillage. Additionally, the code could incorporate feedback mechanisms to adapt to varying bottle sizes or environmental factors. This automation aimed to empower patients in healthcare scenarios, promoting independence and alleviating the burden on healthcare professionals for routine caregiving tasks, enhancing the overall quality of patient care.

3) Deliver food / deliver medicine: Creating a Python code implementation for delivering food or medicine to patients using the UFACTORY xArm7 robotic arm and gripper introduced a transformational aspect to healthcare assistance. The code was designed to facilitate the robotic arm's precise movement to pick up a designated package containing

either food or medicine with the gripper. It then navigated through predefined paths to reach the patient's location and gently release the package. Implementing safety protocols was paramount to ensure the robotic arm operates in a controlled manner, avoiding any harm to patients or surroundings. The code also incorporated obstacle detection and avoidance mechanisms to navigate through dynamic environments. By automating the delivery process, this code aimed to enhance operational efficiency within healthcare facilities, allowing healthcare professionals to focus on more intricate aspects of patient care, while simultaneously providing timely and convenient services to patients in need.

4) Take temperature: Implementing a Python code for the UFACTORY xArm7 robotic arm and gripper to take the temperature of patients in a healthcare setting introduced a sophisticated level of automation. The code instructed the robotic arm to grasp a non-contact infrared thermometer using the gripper, ensuring a secure hold. The arm then carefully approached the patient, positioned the thermometer at the appropriate distance, and triggered the temperature measurement. To enhance the code's functionality, it incorporated features such as patient identification, precise positioning, and data logging for accurate record-keeping. Safety measures, including controlled movement and obstacle detection, were crucial to guarantee the well-being of patients during the temperature-checking process. By automating temperature measurements, this code contributed to the efficiency of healthcare operations, reduced the risk of cross-contamination, and allowed healthcare professionals to allocate more time to critical aspects of patient care.

C. Experimental Performances of Assisted Robotic Arm

The experimental performances of the assisted robot arm were taken as videos. Screen captures of the videos were gathered to show the performance of the assisted arm. Fig. 11 shows assisted robot arm cleaning area, pouring Water, delivering medicine and taking body temperature. Fig. 12 shows the example process of assisted robot arm pouring water.



Fig. 11. Assisted robot Arm cleaning area, pouring water, delivering medicine, taking body temperature.

D. Recorded Robot Movement in 3D modeling Environment of the Assisted Robotic Arm

The UFACTORY *xArm Studio* software provides a three-dimensional modeling environment for recording, visualizing, and replaying robotic movements. During task execution, the positions and orientations of the robotic arm are continuously recorded and displayed using a real-time 3D model that mirrors the physical robot. An example is shown in Fig. 13. This environment allows users to verify joint trajectories, monitor

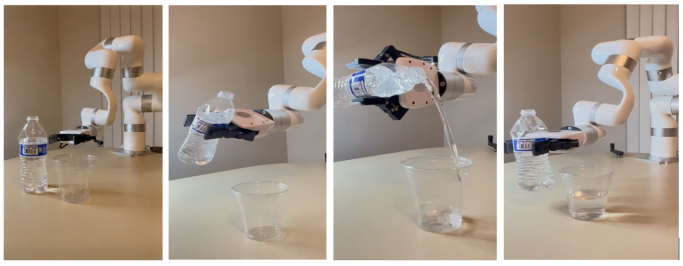


Fig. 12. Assisted robot Arm completing tasks of pouring water.

end-effector motion, and evaluate the sequence of programmed operations before deployment. The recording function also enables previously executed motions to be replayed for debugging, trajectory refinement, and performance validation. By providing a synchronized digital representation of the physical robotic system, the 3D modeling environment facilitates efficient program development and improves the reliability of healthcare assistance tasks.

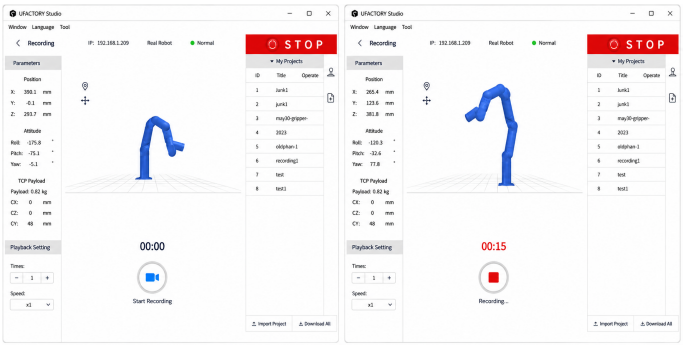


Fig. 13. Recoreded robot movement in 3D modeling environment.

IV. System Validation of Assisted Robotic Arm Performance for Example Tasks and Use Cases

A total of 100 different voice commands related to the supported robotic healthcare tasks were evaluated to assess the performance of the LLM-based task classification system. The test commands represented different natural-language expressions associated with four predefined task categories: `POUR_WATER`, `CLEAN_AREA`, `DELIVER_MEDICINE`, and `TAKE_TEMPERATURE`. Rather than requiring users to speak an exact predefined phrase, the experiment included variations in wording and play the wording with audio to evaluate whether the LLM could identify the intended robotic action from the semantic meaning of each command. Of the 100 commands, 97 were mapped to the intended task category, resulting in an overall classification accuracy of 97%. The three classification errors were associated with commands containing ambiguous or overlapping semantic information. For example, an expression such as "pouring medicine" can contain information related to both the `POUR_WATER` and `DELIVER_MEDICINE` categories. These results indicate that the LLM-based approach can provide flexible natural-language control while also demonstrating the need for ambiguity detection and confirmation mechanisms before physical robot execution.

Table I presents the confusion-matrix representation of the classification results. The diagonal elements indicate correctly classified commands, whereas the off-diagonal elements represent classification errors. In this example, 25 commands were evaluated for each task category, producing 97 correct classifications and three misclassifications.

TABLE I. CONFUSION MATRIX FOR LLM-BASED TASK CLASSIFICATION

Actual Task	Predicted Task			
	Water	**Clean**	**Medicine**	**Temp.**
Water	**24**	0	1	0
Clean	0	**25**	0	0
Medicine	1	0	**24**	0
Temperature	0	0	1	**24**

The three off-diagonal entries represent the three classification errors. The bold diagonal values make the correct classifications particularly easy to identify.

The confusion matrix shows that most predictions are concentrated along the main diagonal, demonstrating strong agreement between the intended task and the task selected by the LLM. The remaining errors illustrate an important limitation of natural-language robotic control: semantically similar or ambiguous expressions can occasionally be assigned to an incorrect action. This issue is particularly important in healthcare-oriented applications because an incorrect classification could initiate an unintended robotic operation. Future versions of the system can therefore incorporate an `UNKNOWN` or `AMBIGUOUS` category, confidence thresholds, and a confirmation step in which the system asks the user to verify the interpreted task before executing potentially safety-sensitive actions. These mechanisms could further improve the reliability of the natural-language interface while preserving the flexibility provided by LLM-based intent interpretation.

The robotic healthcare assistant was validated using repeated task-based experiments. Four representative functions were evaluated: removal of contaminated materials, water pouring, medicine delivery, and body-temperature measurement. Each task was performed ten times under the same experimental conditions. Task success, execution time, end-effector positioning error, repeatability, and operational safety were recorded for every trial.

For motion validation, the commanded Cartesian coordinates were compared with the measured end-effector coordinates. The three-dimensional positioning error was calculated as:

$$E = \sqrt{(X_m - X_t)^2 + (Y_m - Y_t)^2 + (Z_m - Z_t)^2}, \qquad (1)$$

where, (X_t, Y_t, Z_t) represents the commanded target position and (X_m, Y_m, Z_m) represents the measured position. Repeatability was evaluated using the standard deviation of the measured coordinates across repeated trials. A task was considered successfully validated when at least 90% of the trials were completed correctly, the average positioning error remained below 5 mm, and no critical collision, uncontrolled motion, or object-dropping event occurred. The validation results were then compared across the four healthcare tasks to evaluate the reliability of the complete robotic system. The validation criteria for the robot healthcare system is as Table II.

Table III shows the validation details for pouring water. For the pouring-water function specifically, validation test the full sequence from opening the gripper through returning the bottle. The bottle is placed at a fixed starting location, and the cup remains at a known target location. Each trial determine whether the robot successfully grasps the bottle, lifts it without slipping, reaches the pouring position, tilts the bottle, returns it upright, replaces it, and releases it correctly.

TABLE II. VALIDATION CRITERIA FOR THE ROBOTIC HEALTHCARE SYSTEM

Metric	Measurement	Acceptance Criterion
Task Success	Successful trials / total trials	$\geq 90\%$
Position Error	Euclidean distance between target and measured XYZ	≤ 5 mm
Repeatability	Standard deviation of repeated XYZ measurements	≤ 3 mm
Execution Time	Time required to complete each task	Stable across trials
Command Response	Correct execution after user/software command	$\geq 90\%$
Safety	Collision, dropped object, or uncontrolled motion	0 critical events

TABLE III. VALIDATION OF THE ROBOTIC WATER-POURING SEQUENCE

Step	Operation	Expected Result	Criterion	Status
1	Open gripper	Gripper opens	Correct position	Pass
2	Move to bottle	Reaches bottle	Error ≤ 5 mm	Pass
3	Grasp bottle	Bottle secured	No slipping	Pass
4	Lift bottle	Reaches safe height	Error ≤ 5 mm	Pass
5	Move above cup	Reaches pour point	Error ≤ 5 mm	Pass
6	Tilt bottle	Water poured	Correct rotation	Pass
7	Return upright	Pouring stops	No continued flow	Pass
8	Replace bottle	Returns to target	Error ≤ 5 mm	Pass
9	Release bottle	Bottle stable	No tipping	Pass
10	Return arm	Reaches home	Error ≤ 5 mm	Pass

TABLE IV. EXPERIMENTAL VALIDATION RESULTS

Task	Trials	Success	Rate (%)	Time (s)	Error (mm)	Result
Remove Material	10	10	100	10.4	1.42	Pass
Pour Water	10	9	90	10.8	1.61	Pass
Deliver Medicine	10	10	100	9.7	1.28	Pass
Measure Temperature	10	10	100	10.1	1.35	Pass
Overall	**40**	**39**	**97.5**	**10.25**	**1.42**	**Pass**

Table IV presents the experimental validation results of the robotic healthcare assistance system for four representative tasks: removing contaminated material, pouring water, delivering medicine, and measuring body temperature. Each task was repeated ten times to examine the reliability of the robotic arm under consistent experimental conditions. The

system successfully completed 39 of the 40 trials, resulting in an overall task success rate of 97.5%. Three tasks achieved a 100% success rate, while the water-pouring task achieved 90%, indicating one unsuccessful trial. The lower success rate for water pouring can be attributed to the greater manipulation complexity of this task, which requires the robot to grasp the bottle, lift it, move toward the target, control the wrist orientation during pouring, and return the bottle to its original position.

The execution time remained relatively consistent among the four tasks, ranging from 9.7 s for medicine delivery to 10.8 s for water pouring. The overall mean execution time was 10.25 s. Water pouring required the longest time because it involved additional movements for grasping, lifting, tilting, and returning the bottle. In comparison, medicine delivery produced the shortest execution time. These results suggest that the robotic system can complete the selected point-of-care activities within a relatively consistent time range despite differences in task requirements.

Positioning performance was evaluated using the mean error between the target and measured end-effector positions. As shown in Table IV, the mean errors ranged from 1.28 mm to 1.61 mm, with an overall mean of 1.42 mm. Medicine delivery produced the lowest positioning error of 1.28 mm, whereas water pouring produced the highest error of 1.61 mm. The increased error during pouring may result from the additional changes in arm position and wrist orientation required to manipulate the bottle. Nevertheless, all measured errors remained below the predefined 5 mm acceptance threshold. The combination of a 97% overall success rate, an average execution time of 10.25 s, and a mean positioning error of 1.42 mm indicates that the developed robotic arm system can perform the tested healthcare assistance tasks with good repeatability and operational reliability. These results support the feasibility of the proposed system for further development as an intelligent robotic assistant for routine point-of-care applications.

A. Discussion of Errors and Potential Improvements

The experimental results demonstrate that the proposed system can successfully integrate voice recognition, LLM-based intent classification, and robotic task execution; however, several sources of error were observed. For the natural-language control component, the LLM-based task classification achieved an accuracy of 97%, indicating that a small proportion of the tested commands were incorrectly classified. One important source of error was ambiguity in the user's spoken request. For example, an expression such as "pouring medicine" contains semantic information associated with both the POUR_WATER and DELIVER_MEDICINE classes. Consequently, the LLM may assign an incorrect task when the input contains overlapping terminology or does not clearly describe the intended action. Speech-to-text errors can further contribute to incorrect classifications because inaccurate transcription changes the textual input provided to the LLM.

Several approaches can be used to improve the reliability of the natural-language control component. First, an `UNKNOWN` class can be incorporated so that ambiguous or low-confidence commands do not immediately initiate robotic motion. Instead, the system can request clarification from the user, such as asking whether the user intends to pour water or receive medicine. Second, task-specific prompts and additional representative command examples can be provided to the LLM to improve intent classification. The system can also require confirmation before executing safety-sensitive activities, particularly medicine-related tasks. These modifications would reduce the probability that an incorrectly interpreted command results in an unintended robotic action.

Errors can also occur during physical robotic execution. Although predefined Cartesian coordinates allow the xArm7 to repeatedly approach known target locations, small positioning differences can result from calibration uncertainty, coordinate-system alignment, object-placement variation, gripper positioning, and mechanical repeatability. Manipulation tasks such as grasping a bottle and pouring water are particularly sensitive because a small displacement of the bottle or cup can affect grasping and pouring performance. Differences between commanded and measured end-effector coordinates therefore provide an important quantitative indication of robotic positioning error.

Currently the level of intelligence of automation is at the basic level where the motion were predefined without sensor input to enhance the level of intelligence. Future improvements should incorporate additional sensor feedback rather than relying primarily on predefined positions and motion sequences. A camera-based vision system could identify the real-time locations of bottles, cups, medicine containers, and other objects before manipulation. Visual feedback could then be used to dynamically modify the target coordinates instead of assuming that objects remain at exactly the same locations. Force or tactile sensing at the gripper could additionally determine whether an object has been successfully grasped and detect unexpected contact. Combining visual perception, force feedback, collision detection, and closed-loop trajectory correction would enable the robotic system to adapt to variations in the environment and improve manipulation reliability.

Further improvements also include adding more trials for the validation. For example, for the validation of the LLMs, we didn't test on the execution time and potential delays although through observation, those were not the major potential problems for the system to function. Moreover, for the assisted robot arm's performance, the 97.5 percent overall success rate and a 1.42 mm mean positioning error, rest on only ten trials per task, so the work can be toward reporting trial-level variability or increase the sample size. More comparisons could be done to compare with previous studies in the same scope.

Overall, the observed errors highlight the importance of validating both the intelligent interaction layer and the physical robotic layer. The LLM provides flexibility in understanding natural-language requests, while predefined robot routines provide controlled task execution; nevertheless, uncertainty can be introduced at the speech-recognition, intent-classification, coordinate-calibration, and manipulation stages. Future development should therefore emphasize confidence-based intent classification, user confirmation for ambiguous commands, improved calibration, real-time object detection, sensor-based feedback, and closed-loop robotic control. These improvements could increase the robustness of the proposed system

and provide a stronger foundation for future evaluation in more realistic healthcare environments.

V. Discussions

The findings of this study demonstrate the feasibility of employing a collaborative robotic manipulator as an intelligent assistant for healthcare applications. The proposed system successfully integrates robotic manipulation, autonomous task execution, and software-based control into a unified platform capable of supporting routine patient-care activities. By automating repetitive bedside tasks, such as object delivery and physiological monitoring, the robotic assistant has the potential to reduce the physical workload of healthcare professionals while allowing them to devote greater attention to clinical decision-making and direct patient interaction. The integration of collaborative robotics into healthcare environments therefore represents a promising approach for improving operational efficiency without replacing the essential role of medical personnel.

One of the primary contributions of this work is the development of a modular software architecture that combines robot motion control, task planning, user interaction, and potential sensor integration using Python. The modular design enables individual system components to be independently updated or expanded, facilitating future integration of additional sensing devices and artificial intelligence techniques.

The practical deployment of collaborative robotic assistants requires consideration of economic and operational factors. As collaborative robot technology continues to mature and hardware costs decrease, robotic assistants may become increasingly cost-effective, particularly for performing predictable and repetitive tasks that consume substantial nursing resources. Improvements in usability, interoperability with hospital information systems, and standardized deployment procedures are planned for future work.

VI. Conclusions and Future Work

This study developed and evaluated an intelligent robotic patient-assistance framework that integrates collaborative robotic manipulation, voice-based natural-language interaction, and non-contact physiological monitoring. The UFACTORY xArm7 collaborative manipulator was used as the primary robotic platform to perform representative patient-care tasks, including cleaning a designated area, pouring water, delivering medicine, and taking body temperature. The proposed framework demonstrates how a collaborative robotic arm can be programmed to provide physical assistance for predictable and repetitive bedside activities while maintaining a software architecture that can be expanded to accommodate additional healthcare-related functions.

A significant contribution of this research is the integration of natural-language voice control with robotic task execution. The developed software uses continuous speech-to-text processing to convert spoken user requests into textual information, which is subsequently analyzed by a Large Language Model to determine the intended robotic task. This approach enables users to communicate with the robotic assistant using natural expressions rather than memorizing a fixed set of commands. Evaluation demonstrates the potential of LLM-based intent recognition for intuitive human–robot interaction. However, ambiguous expressions, such as commands containing terminology associated with multiple tasks, can result in incorrect classifications.

The integration of non-contact physiological monitoring further extends the functionality of the robotic platform beyond conventional object manipulation. In this study, body-temperature measurement was incorporated as a representative monitoring task in which the robotic arm positions a non-contact infrared thermometer for patient measurement. Combining physical assistance, physiological sensing, and natural-language interaction within the same robotic framework provides a foundation for developing more comprehensive patient-support systems. Future implementations could extend this architecture to additional non-contact sensors and more advanced perception capabilities while preserving appropriate clinical supervision.

Overall, the findings demonstrate the feasibility of combining collaborative robotics, Python-based robot control, LLM-enabled natural-language interaction, and physiological sensing within a unified healthcare-assistance framework. The proposed system is not intended to replace nurses or other healthcare professionals; instead, it provides a technological approach for supporting selected repetitive and predictable activities that can consume caregiver time. Future research should focus on improving recognition of ambiguous commands, integrating environmental perception and obstacle avoidance, expanding physiological sensing capabilities, and conducting evaluations in more realistic patient-care environments. Further development of these capabilities could enable robotic assistants to provide more adaptive and accessible support while allowing healthcare professionals to concentrate on clinical assessment, decision-making, and direct patient interaction.

Beyond healthcare assistance, the proposed collaborative robotic platform has potential to support individuals with upper-limb impairments in performing activities of daily living. A collaborative robotic arm positioned beside a bed or wheelchair can assist with tasks such as retrieving personal items, delivering food or medication, and providing drinking assistance through intuitive user interfaces, including touch or voice commands. By combining autonomous manipulation with intelligent motion planning, the robotic system can improve user independence, reduce reliance on caregivers, and enhance quality of life. Future research may incorporate advanced perception and artificial intelligence to provide more adaptive, personalized assistance in home and rehabilitation environments.

References

[1] World Health Organization. *State of the World's Nursing 2025: Investing in Education, Jobs, Leadership and Service Delivery*; World Health Organization: Geneva, Switzerland, 2025.

[2] L. Z. Li, L. X. Yang, L. J. Liu, and F. F. Wu, "Nurse burnout and patient safety, satisfaction, and quality of care: A systematic review and meta-analysis," *JAMA Network Open*, vol. 7, no. 11, p. e2443059, 2024. doi: 10.1001/jamanetworkopen.2024.43059.

[3] "Contributing factors to missed nursing care: A systematic review with clinical implications for practice and patient safety," *Applied Nursing Research*, vol. 85, p. 152002, 2025. doi: 10.1016/j.apnr.2025.152002.

[4] X. Meng, X. Yan, K. Zhang, *et al.*, "The application of large language models in medicine: A scoping review," *iScience*, vol. 27, no. 5, Art. no. 109713, 2024. doi: 10.1016/j.isci.2024.109713.

[5] F. Busch, L. Hoffmann, C. Rueger, *et al.*, "Current applications and challenges in large language models for patient care: A systematic review," *Communications Medicine*, vol. 5, Art. no. 26, 2025.

[6] Ball, J.E.; Bruyneel, L.; Aiken, L.H.; Sermeus, W.; Sloane, D.M.; Rafferty, A.M.; Lindqvist, R.; Tishelman, C.; Griffiths, P. Post-operative mortality, missed care and nurse staffing in nine countries: A cross-sectional study. *Int. J. Nurs. Stud.* **2018**, *78*, 10–15.

[7] Ohneberg, C.; Stöbich, N.; Warmbein, A.; Rathgeber, I.; Mehler-Klamt, A.C.; Fischer, U.; Eberl, I. Assistive robotic systems in nursing care: A scoping review. *BMC Nurs.* **2023**, *22*, 72.

[8] Bemelmans, R.; Gelderblom, G.J.; Jonker, P.; de Witte, L. Socially assistive robots in elderly care: A systematic review into effects and effectiveness. *J. Am. Med. Dir. Assoc.* **2012**, *13*, 114–120.e1.

[9] Broadbent, E.; Stafford, R.; MacDonald, B. Acceptance of healthcare robots for the older population: Review and future directions. *Int. J. Soc. Robot.* **2009**, *1*, 319–330.

[10] Papadopoulos, I.; Koulouglioti, C.; Ali, S. Views of nurses and other health and social care workers on the use of assistive humanoid and animal-like robots in health and social care: A scoping review. *Contemp. Nurse* **2020**, *56*, 102–119.

[11] Feil-Seifer, D.; Matarić, M.J. Defining socially assistive robotics. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics*; Chicago, IL, USA, 2005; pp. 465–468.

[12] Goodrich, M.A.; Schultz, A.C. Human–robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.* **2008**, *1*, 203–275.

[13] Bhattacharjee, T.; Gordon, E.K.; Srinivasa, S.S. Assistive robot manipulation for activities of daily living. *Annu. Rev. Control Robot. Auton. Syst.* **2020**, *3*, 1–27.

[14] Park, D.; Hoshi, Y.; Mahajan, H.P.; Kim, H.K.; Erickson, Z.; Rogers, W.A.; Kemp, C.C. Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. *Robot. Auton. Syst.* **2020**, *124*, 103344.

[15] Siegel, J.D.; Rhinehart, E.; Jackson, M.; Chiarello, L. *2007 Guideline for Isolation Precautions: Preventing Transmission of Infectious Agents in Healthcare Settings*; Centers for Disease Control and Prevention: Atlanta, GA, USA, 2007.

[16] Nelson, A.; Baptiste, A.S. Evidence-based practices for safe patient handling and movement. *Online J. Issues Nurs.* **2006**, *11*, 3.

[17] Davis, K.G.; Kotowski, S.E. Prevalence of musculoskeletal disorders for nurses in hospitals, long-term care facilities, and home health care: A comprehensive review. *Hum. Factors* **2015**, *57*, 754–792.

[18] Barr, P.; Zhang, Q.; Rostami, M.R. Protecting healthcare workers during infectious disease outbreaks: Occupational exposure and control strategies. *J. Hosp. Infect.* **2020**, *105*, 608–615.

[19] Mutlu, B.; Roy, N.; Forlizzi, J.; Hodgins, J. Robots in healthcare: Designing collaborative robotic assistants for clinical environments. *Annu. Rev. Biomed. Eng.* **2020**, *22*, 339–364.

[20] Dario, P.; Guglielmelli, E.; Allotta, B.; Carrozza, M.C. Robotics for Medical Applications. *IEEE Robot. Autom. Mag.* **2000**, *7*, 44–56.

[21] Kragic, D.; Christensen, H.I. Survey on Visual Servoing for Manipulation. *Comput. Vis. Image Underst.* **2009**, *113*, 1–16.

[22] Tsui, K.M.; Yanco, H.A.; Cramer, H.; Kemper, N. Using Autonomous Robots to Support Healthcare Workers in Hospitals. In *Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction (HRI)*; Lausanne, Switzerland, 2011; pp. 209–216.

[23] Javdani, S.; Bagnell, J.A.; Srinivasa, S.S. Shared Autonomy via Hindsight Optimization. *Int. J. Robot. Res.* **2018**, *37*, 717–742.

[24] Kim, B.; Pineau, J. Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning. *Int. J. Soc. Robot.* **2016**, *8*, 51–66.

[25] Maheu, V.; Frappier, J.; Archambault, P.; Routhier, F. Evaluation of the JACO Robotic Arm: Clinico-Economic Study for People with Upper-Extremity Disabilities. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*; Zurich, Switzerland, 2011.

[26] Heerink, M.; Kröse, B.; Evers, V.; Wielinga, B. Assessing Acceptance of Assistive Social Robots by Older Adults: The Almere Model. *Int. J. Soc. Robot.* **2010**, *2*, 361–375.

[27] Beer, J.M.; Smarr, C.A.; Chen, T.L.; Prakash, A.; Mitzner, T.L.; Kemp, C.C.; Rogers, W.A. The Domesticated Robot: Design Guidelines for Assisting Older Adults to Age in Place. In *Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction (HRI)*; Boston, MA, USA, 2012; pp. 335–342.

[28] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, pp. 1930–1940, 2023. doi: 10.1038/s41591-023-02448-8.

[29] J. N. Kather, D. Ferber, I. C. Wiest, S. Gilbert, and D. Truhn, "Large language models could make natural language again the universal interface of healthcare," *Nature Medicine*, vol. 30, pp. 2708–2710, 2024. doi: 10.1038/s41591-024-03199-w.

[30] D. Wang and S. Zhang, "Large language models in medical and healthcare fields: Applications, advances, and challenges," *Artificial Intelligence Review*, vol. 57, Art. no. 299, 2024. doi: 10.1007/s10462-024-10921-0.

[31] K. Singhal, S. Azizi, T. Tu, *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, pp. 172–180, 2023.

[32] UFACTORY, *xArm User Manual*. Shenzhen, China: UFACTORY Robotics, 2025.

[33] UFACTORY, *xArm Studio User Guide*. Shenzhen, China: UFACTORY Robotics, 2025.

[34] UFACTORY, *xArm Developer Documentation*. Shenzhen, China: UFACTORY Robotics, 2025. [Online]. Available: https://docs.xarm.ufactory.cc

[35] Intel Corporation, "Intel Core i7-13700F processor specifications," [Online]. Available: https://www.intel.com/content/www/us/en/products/sku/230490/intel-core-i713700f-processor-30m-cache-up-to-5-20-ghz/specifications.html. [Accessed: Aug. 3, 2026].

[36] NVIDIA Corporation, "GeForce RTX 4070 specifications," [Online]. Available: https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4070/. [Accessed: Aug. 3, 2026].

[37] Micro-Star International (MSI), "MSI gaming desktop products," [Online]. Available: https://www.msi.com/Desktop. [Accessed: Aug. 3, 2026].

[38] Microsoft, "Windows 11 specifications and system requirements," [Online]. Available: https://www.microsoft.com/windows/windows-11-specifications. [Accessed: Aug. 3, 2026].

When Agents Delegate: Accountability Chains in Multi-Agent AI Systems

Ornella Bahidika
Independent Researcher, Seattle, USA

***Abstract*—Ethical frameworks for the oversight of autonomous AI systems generally assume a dyadic relationship: one human principal authorizing and supervising one artificial agent. Deployed practice has moved on. Contemporary agentic systems decompose tasks and delegate them onward, to sub-agents they instantiate, to third-party agents discovered through interoperability protocols, and to tools whose own behavior is agentic. Each hop in such a delegation chain attenuates the three prerequisites on which accountability depends: authorized scope, attributable intent, and reconstructable provenance. This study analyzes the resulting accountability gap. We contribute a taxonomy of delegation structures in deployed agentic systems; an analysis of four failure modes by which delegation dissolves accountability, which we term authorization dilution, intent drift, provenance loss, and responsibility laundering; a normative framework, the Delegation Accountability Chain, comprising five conditions under which multi-agent delegation remains compatible with meaningful human oversight; an audit of five widely used interoperability protocols and orchestration frameworks (the Model Context Protocol, the Agent2Agent protocol, LangGraph, AutoGen, and the OpenAI Agents SDK) against the five conditions, finding that none satisfies any condition chain-wide at a mandatory level; and a mapping of the conditions onto the human-oversight obligations of the EU Artificial Intelligence Act and the NIST AI Risk Management Framework, both drafted for the dyadic case. We argue that the question of where the buck stops has a defensible answer in multi-agent systems only if delegation is constrained by design.**

Keywords*—*Agentic AI; accountability; delegation; multi-agent systems; human oversight; AI governance; responsibility gap

I. Introduction

In earlier work we asked where the buck stops when an autonomous AI system acts consequentially on a human's behalf, proposed an action-level tiered oversight framework calibrated by stakes and reversibility, and identified tier laundering, the composition of individually low-tier actions into a higher-tier outcome, as the central methodological gap in per-action oversight [1]. That work named composition-aware oversight as the highest-leverage direction for follow-up research. Like nearly all of the human-oversight literature it engaged, that earlier analysis presumed a single agent answerable to a single accountable human. The present study takes up its follow-up program for the setting in which composition is most acute and least governed: delegation among agents.

The deployed world has moved past that presumption. Production agentic systems now plan by decomposition: an orchestrating agent breaks a goal into subtasks and assigns them to other agents. Interoperability protocols such as the Model Context Protocol and agent-to-agent communication standards make it routine for an agent to enlist capabilities that its human principal never inspected, sourced from parties the principal never chose [2], [3]. Each such hop is a delegation, an operation classical accountability theory treats with suspicion: Bovens defines accountability as a relationship in which an actor is obliged to explain and justify conduct to a forum that can pose questions and impose consequences [4]. A forum cannot pose questions to a chain whose links it cannot see. Fig. 1 contrasts the two settings.

This study asks what oversight can mean when the overseen agent is itself a principal to other agents. We proceed as follows. Section II situates the problem in the responsibility-gap literature and shows that current regulatory instruments assume the dyadic case. Section III offers a taxonomy of delegation structures observed in deployed systems. Section IV analyzes four failure modes by which delegation dissolves accountability. Section V presents the Delegation Accountability Chain, five conditions under which delegation remains compatible with meaningful human oversight. Section VI audits five widely deployed interoperability protocols and orchestration frameworks against the conditions. Section VII maps the conditions onto the EU AI Act and the NIST AI RMF. Section VIII considers objections and limitations, and Section IX concludes.

II. Background and Related Work

Three bodies of work frame human oversight of AI, and all three exhibit the dyadic assumption.

1) Responsibility gaps: Matthias observed two decades ago that learning machines create situations in which no human satisfies the classical conditions for responsibility, control and knowledge [5]. Nissenbaum earlier identified "many hands" as a barrier to accountability in computerized systems [6]. Delegating agents combine both problems: the hands are now artificial and generated at runtime.

2) Meaningful human control: Santoni de Sio and van den Hoven's influential account requires that autonomous systems track the reasons of the relevant humans and that their behavior be traceable to a proper moral and technical understanding by designers and operators [7]. Both the tracking and tracing conditions are defined against a fixed system boundary. Delegation makes the boundary dynamic: the "system" at time t includes agents that did not exist at deployment time.

3) Multi-agent systems and capability security: Two older literatures bear directly on delegation and are rarely joined to the oversight debate. Multi-agent systems research has studied cooperation, negotiation, and delegation among software

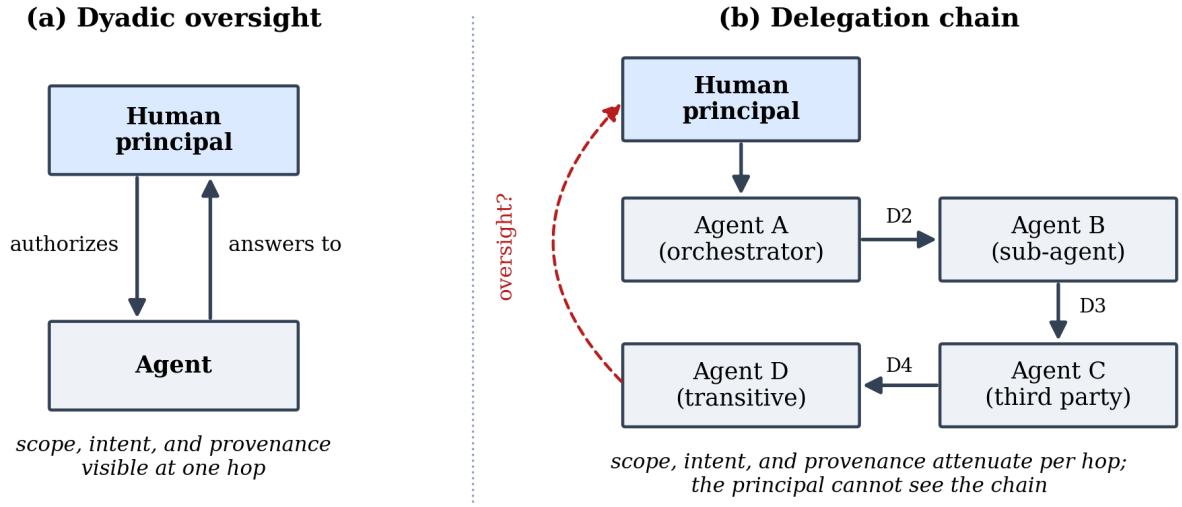


Fig. 1. Dyadic oversight versus a delegation chain. In the dyad, authorization and answerability meet at a single hop. In a chain, scope, intent, and provenance attenuate per hop, and the principal cannot see the chain acting under its grant.

agents since the 1990s [8], but its concern was coordination among agents, not accountability of agents to humans; the field's delegation formalisms assume the principals are themselves agents. Capability security, by contrast, is about conveying bounded authority: Hardy's confused-deputy analysis showed that authority which travels implicitly is authority that gets misused [9], and the object-capability tradition demonstrated that authorization can be made to travel as an unforgeable artifact that composes safely [10]. The framework of Section V can be read as importing the capability discipline into the human-oversight debate: what capability systems do for programs, delegation-accountability conditions must do for chains that terminate in a responsible person.

4) Regulatory instruments: Article 14 of the EU AI Act requires that high-risk systems be designed so that natural persons can effectively oversee them, including the ability to understand capacities and limitations, remain aware of automation bias, correctly interpret output, and intervene or interrupt [11]. The NIST AI Risk Management Framework likewise organizes its Govern, Map, Measure, and Manage functions around an identified system with an identified operator [12]. Neither instrument contemplates a system that recruits other systems after deployment; both implicitly assume the object of oversight is the object that was assessed.

Recent work has begun to note the strain. Ethics-based auditing programs presume access to a bounded artifact [13]; scholars of the algorithmic society observe that autonomy blurs attribution [14]; practitioner commentary asks the buck-stops question of agentic AI directly [15]; and Chan et al. propose concrete visibility mechanisms for agents, including identifiers, real-time monitoring, and activity logs [16]. Closer still, concurrent work proposes point mechanisms for delegation itself: a cryptographic protocol that accumulates signed per-hop records anchored in a human principal's declared scope [17], identity-governance infrastructure for propagating authorization through multi-agent systems [18], and a draft Internet standard for human-anchored agent identity, delegation, and provenance [19]. These proposals supply individual mechanisms. What this study adds is an account of delegation as the mechanism by which attribution dissolves; an argument that the five conditions of Section V must hold jointly and be defined over chains rather than agents, since a compliant agent delegating to a compliant agent does not yield a compliant chain; and an audit showing that none of these mechanisms, proposed or shipped, is required by the specifications agents actually run on.

III. A Taxonomy of Delegation in Deployed Agentic Systems

We distinguish four delegation structures, ordered by increasing accountability risk, and note a deployed public example of each.

1) D1. Static tool invocation: The agent calls a fixed, enumerated tool (a search API, a database query). The tool set is known at deployment and can be assessed with the system. This is delegation only in a degenerate sense; classical oversight frameworks handle it. Mainstream chat assistants invoking a fixed web-search or code-execution tool exemplify this class.

2) D2. Self-decomposition: An orchestrating agent instantiates subordinate agents of its own kind, assigns them subtasks, and integrates their results. The sub-agents inherit the orchestrator's capabilities but not necessarily its constraints; whether instructions, permissions, and prohibitions survive the handoff is an implementation choice, often an accidental one. Anthropic's production research system is a documented public example: a lead orchestrator agent decomposes a query and spawns parallel subagents whose results it synthesizes [20].

3) D3. Heterogeneous enlistment: The agent discovers and engages third-party agents or services through an interoperability protocol. The enlisted party was built by a different organization, is governed by different policies, and may change its behavior between the principal's assessment and the moment of use. The MCP ecosystem exemplifies this class: hosts routinely connect to community-published third-party servers,

and the A2A protocol was designed expressly for cross-vendor agent enlistment [2], [3].

4) D4. Transitive delegation: An enlisted agent itself delegates onward, producing chains of depth greater than two. No party in the chain, including the original human principal, has visibility over the full set of actors, and depth is unbounded in current protocols. The audited specifications themselves contemplate this class: an MCP server may act as an OAuth client to upstream services, and A2A permits a server to relay authorization requirements through a chain of dependent tasks (Section VI).

Table I summarizes the four classes.

TABLE I. DELEGATION STRUCTURES IN DEPLOYED AGENTIC SYSTEMS

Class	Delegate set	Public example	Principal's visibility
D1 Static tool invocation	Fixed and enumerable at deployment	Chat assistant with a fixed search or code tool	Full
D2 Self-decomposition	Instantiated at runtime by the orchestrator	Orchestrator spawning parallel subagents [20]	Partial
D3 Heterogeneous enlistment	Discovered at runtime from third parties	Community MCP servers; A2A cross-vendor agents [2], [3]	Low
D4 Transitive delegation	Recursively assembled, unbounded	Servers acting as clients to upstream services (Section VI)	None

Two properties make D2 through D4 categorically different from D1. First, the delegate set is determined at runtime, so pre-deployment assessment cannot enumerate it. Second, delegation is composable, so any framework that governs only single hops fails by induction: a compliant agent delegating to a compliant agent does not yield a compliant chain unless compliance is defined over chains.

IV. FOUR WAYS DELEGATION DISSOLVES ACCOUNTABILITY

A. Authorization Dilution

A human authorizes an agent for a scope: purposes, resources, and limits. When the agent delegates, the effective scope of the delegate is whatever the delegating agent communicates, which may be broader than, narrower than, or simply different from the original grant. Across hops, the human's authorization becomes a rumor. The delegate acts under a scope no human issued. Where a delegate exercises inherited credentials broader than the human's grant, this is the confused-deputy problem in its classic ambient-authority form [9]; where scope degrades through linguistic restatement, the failure is distinct but the remedy is the same: authority must travel explicitly or not at all.

B. Intent Drift

Each delegation restates the goal. Restatement is interpretation, and interpretation by a language model is lossy and generative at once: constraints are dropped, emphases shift, and instrumental subgoals are elevated to ends. The terminal agent in a chain may faithfully execute an instruction that no human ever gave. Drift is not misbehavior by any single link; every link may act reasonably on what it received. The failure is emergent, which is why per-agent oversight cannot detect it.

C. Provenance Loss

Accountability requires reconstruction: which information the system received, what it retrieved, which actions it took, and why [21]. In a delegation chain, the records that would answer these questions are scattered across parties with different logging practices, retention policies, and disclosure obligations, or are not produced at all. The forum, in Bovens's sense, has no dossier to examine. Post-incident, it can be impossible even to establish that a chain existed.

D. Responsibility Laundering

The most corrosive failure is intentional or negligent exploitation of the first three: structuring action through delegation so that no identifiable party bears responsibility. The pattern is kin to agency laundering, in which humans obscure their own responsibility by routing decisions through automated systems [22]; responsibility laundering is its multi-agent completion: the routing is performed by agents, and no single human need intend the obfuscation. It also generalizes the tier-laundering pattern identified in our earlier framework [1]: where tier laundering composes low-tier actions within a single agent into a high-tier outcome, responsibility laundering composes individually oversightable agents into an unoversightable chain. Where a human organization performing the same decomposition would face joint liability doctrines, agent chains today face none. If accountability attaches only to single human-agent dyads, delegation becomes a machine for converting responsible action into unattributable outcome. Regulation that ignores this creates an incentive gradient toward deeper chains.

E. A Worked Example

The four failure modes rarely occur alone; a single ordinary chain can exhibit all of them. Consider a procurement assistant, authorized by an operations manager to renew existing software subscriptions up to a set monthly amount. Faced with a renewal that requires a vendor form, the orchestrator spawns a sub-agent to handle vendor interaction (self-decomposition) and, finding the vendor exposes an agent interface, enlists it through an interoperability protocol (heterogeneous enlistment). The vendor's agent, in turn, calls its own payment-processing agent (transitive delegation). At the first hop, the manager's spending cap and renewals-only restriction survive only as sentences in a prompt; the sub-agent inherits the orchestrator's payment credential, whose scope is whatever the credential allows rather than what the manager authorized (authorization dilution). Restating the goal as "complete the vendor's renewal process" drops the cap and the renewals-only qualifier, and the vendor's agent reasonably treats an upsell bundle as part of that process (intent drift). When the resulting charge exceeds anything the manager approved, the reconstruction spans the deployer's logs, the vendor's logs, and a payment processor's records, none linked by a shared identifier and none obliged to disclose (provenance loss). Each party can truthfully say it executed the instructions it received, and the vendor can note that its agent was never told about

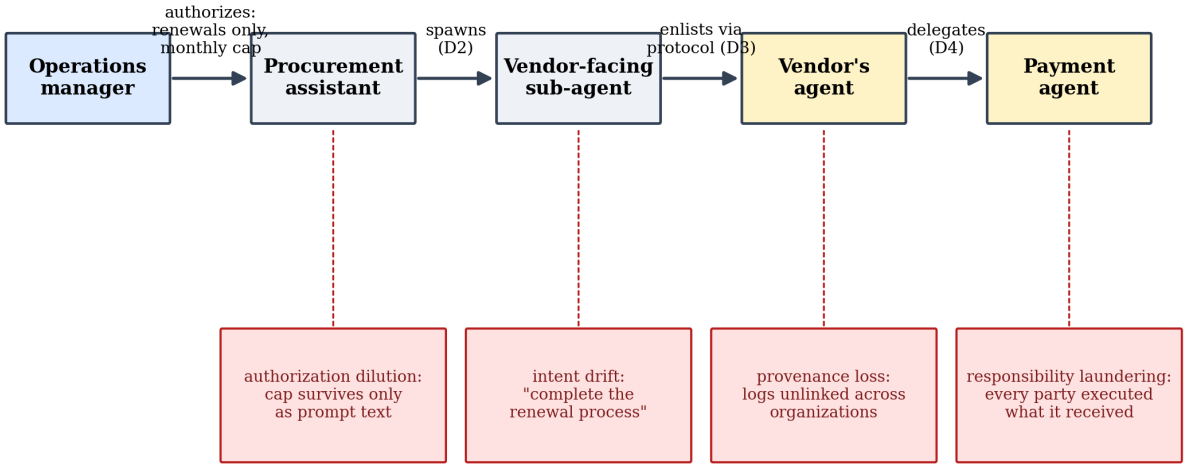


Fig. 2. The worked example: an ordinary procurement chain exhibiting all four failure modes. No party misbehaves; the accountability failure is structural.

any cap; no identifiable party bears the failure (responsibility laundering). Nothing in this chain requires malice or unusual carelessness. Fig. 2 traces the chain and its failures. Section VI shows that the infrastructure such a chain would run on today provides none of the mechanisms that would have interrupted it.

V. The Delegation Accountability Chain

We propose five conditions, jointly necessary for a delegating agentic system to support claims of meaningful human oversight. The conditions are design constraints: each is checkable against an implementation.

1) C1. Conserved authorization: A delegate's effective scope must be a subset of the delegator's scope, enforced mechanically rather than communicated linguistically. Scope must travel as a verifiable artifact (a capability token or its equivalent) that downstream agents cannot expand. The chain-wide invariant is that no action occurs outside the original human grant.

2) C2. Human anchor: Every chain terminates, upward, in an identified natural person or accountable body that issued the originating authorization and remains the answerable party for the chain's aggregate conduct. Anchorship is non-delegable: an agent cannot become an anchor. This preserves Bovens's forum relationship: there is always someone obliged to explain.

3) C3. Attributable initiation: Every delegation event must record who delegated, to whom, under what communicated goal, and under what scope, in a form attributable to the anchor's chain. Delegation to parties that refuse attribution is prohibited conduct, not a logging gap.

4) C4. Chain-complete provenance: The reconstruction standard that audit scholarship applies to single systems [21] must be defined over the chain: a single query, anchored at the human principal, must be able to produce the full delegation tree with per-link records sufficient to answer what was received, retrieved, and done, and what rationale was stated. Practically this requires provenance to propagate with the work, not to be assembled forensically afterward.

5) C5. Bounded depth and diameter: Because drift and dilution compound per hop, chains must carry explicit limits on delegation depth and on the set of admissible delegates, set by the anchor as part of the original grant. Unbounded delegation defeats oversight, and assessors should treat it as a per se failure of Article 14-style obligations.

Each condition carries an operational test an assessor can apply; Table II states them.

TABLE II. The Delegation Accountability Chain: Conditions and Operational Tests

Condition	Requirement	Operational test
C1 Conserved authorization	Delegate scope is a mechanically enforced subset of the delegator's	Every delegated credential is cryptographically derived from, and attenuates, the parent grant
C2 Human anchor	Every chain terminates upward in an identified accountable person or body	Any action in the chain resolves in a single lookup to an identified anchor
C3 Attributable initiation	Every delegation event is recorded and attributable	Every delegation edge exists as a typed record naming delegator, delegatee, goal, and scope
C4 Chain-complete provenance	The full delegation tree is reconstructable from the origin	One anchor-issued query returns the full tree with per-link records
C5 Bounded depth and diameter	The grant limits delegation depth and admissible delegates	The grant carries explicit bounds the runtime refuses to exceed

Deterministic enforcement is the connective tissue of the five conditions. Linguistic instruction is the medium of drift; only mechanisms outside the model, gating code that validates scope, records provenance, and refuses noncompliant delegation, can carry authorization across hops without loss. The design stance is that the agent proposes and the mechanism decides, a pattern practitioners have converged on independently for reliability reasons, and which we here argue has normative status: it is what makes delegation oversightable at all.

VI. Do Current Protocols Satisfy the Chain Conditions? An Audit

The conditions of Section V would be of limited interest if the ecosystem already met them. To assess this, we audited the current normative specifications and official documentation of five artifacts, selected as the most widely adopted interoperability protocols and multi-agent orchestration frameworks at the time of writing (Table III): the Model Context Protocol (MCP, specification revision 2026-07-28), the Agent2Agent protocol (A2A, specification 1.0.0), and the LangGraph, Microsoft AutoGen, and OpenAI Agents SDK frameworks (documentation as of August 2026). For each artifact we assessed each condition as satisfied (a mandatory, MUST-level mechanism exists at protocol or framework level), partial (optional, SHOULD-level, or achievable only through application-level configuration), or absent (no relevant provision; we record what was searched and not found). Two clarifications govern the exercise. First, the three frameworks are developer libraries that make no governance claims of their own; we audit them not against their own claims but against the oversight claims deployers make for systems built on them, because those are the claims the conditions of Section V underwrite. Second, the mandatory-mechanism bar is deliberate. For accountability in particular, optional recording is weak assurance, because the parties best positioned to disable a record are the parties it would hold to account (Section IV-D); default-on but disable-able is therefore materially different from mandatory. We audit what each artifact guarantees rather than what a diligent application could build atop it; whatever is not guaranteed is left to the discretion of each implementer. Audit working notes, including full verbatim quotations and source URLs for every assessment, are available as supplementary material (S1: audit working notes; S2: machine-readable audit evidence) at https://github.com/ornellabahidika/when-agents-delegate-supplement.

No cell in the table is satisfied; Fig. 3 visualizes the verdicts. We discuss each artifact, then the pattern.

A. Model Context Protocol

MCP contains the strongest single-hop authorization mechanics of any audited artifact, and their limits are instructive. When authorization is used, it is mandatory in the details: clients MUST implement OAuth 2.0 Resource Indicators (RFC 8707), servers "MUST validate that access tokens were issued specifically for them as the intended audience," and token passthrough is prohibited ("MCP servers MUST NOT accept or transit any other tokens"); the protocol's separate security-best-practices document repeats the ban ("MCP servers MUST NOT accept any tokens that were not explicitly issued for the MCP server"). But authorization as a whole is "OPTIONAL for MCP implementations," applies only to HTTP transports (local stdio servers sit outside the framework and are directed to ambient environment credentials, the classic confused-deputy configuration [9]), and it is pairwise: each hop obtains a separate token from its own authorization server, so a delegate's effective scope is set by that server's policy, not derived from the delegator's grant; scope subsetting appears only as guidance ("MCP clients SHOULD follow the principle of least privilege"). The consent model is anchored at the human-to-host hop and the specification is direct about its reach: "While MCP itself cannot enforce these security principles at the protocol level," implementors are asked, at SHOULD strength, to build robust consent and authorization flows. No message or field records a delegation event (C3), and the spec's own security guidance concedes that passthrough patterns "make incident investigation, controls, and auditing more difficult" without requiring any audit record in response. The architecture works against chain reconstruction by design: servers "should not be able to read the whole conversation, nor 'see into' other servers" (C4). No provision bounds chain length or membership; composability is a design goal ("Servers should be highly composable"), and a server may itself act as an OAuth client to upstream services (C5).

B. Agent2Agent Protocol

A2A mandates per-request security: "Servers MUST reject requests with invalid or missing authentication credentials" and "Servers MUST implement authorization checks on every A2A Protocol Operations request." An OpenID Connect security scheme is defined, so a human principal can be authenticated at a given hop, though the authenticated client is typically an agent or service; the scheme carries end-user identity only where a deployment arranges it. But identity does not travel: "A2A protocol payloads… don't carry user or client identity information directly. Identity is established at the transport/HTTP layer," which means the originating human is invisible one hop beyond the client (C2 partial). No requirement records who delegated what to whom; the protocol's contextId and referenceTaskIds group related tasks informationally but are not delegation records (C3 absent). Tracing and audit appear only in non-normative enterprise guidance at SHOULD strength (C4 partial). Notably, A2A does define one chain-aware construct: a task needing credentials can propagate that need upstream through a chain of tasks in an authorization-required state, with the sensible rule that a credential obtained this way "MUST NOT be assumed to authorize subsequent messages on the Task." This is a credential-acquisition relay rather than an accountability mechanism, but it demonstrates that chain-scoped constructs are expressible at protocol level when designers choose to define them. Nothing bounds chain depth, and agent discovery through published AgentCards is open-ended (C5 absent).

C. LangGraph

LangGraph's multi-agent documentation describes handoffs as control transfer: "Agents transfer control to each other via tool calls." What travels is graph state and control flow, not authorization; a subordinate node's tools are whatever the developer wired in, with no mechanical relation to the delegator's scope (C1 absent). Human involvement exists as the interrupt() primitive, which can "pause before a critical action and ask for approval," but it is optional, per-node, and identifies no one: the docs speak of a generic "human" with no identity or authentication attached (C2 partial). Recording is likewise optional: the docs recommend tracing "the full coordination flow across agents with LangSmith," an external opt-in service, and persistent state history requires configuring a checkpointer (C3, C4 partial). The framework's one always-on limit is enforced: a default recursion limit of 1,000 steps, raised from the long-standing default of 25 in version 1.0.6, with a

TABLE III. AUDIT RESULTS (SATISFIED / PARTIAL / ABSENT)

Artifact	C1 conserved authorization	C2 human anchor	C3 attributable initiation	C4 chain provenance	C5 bounded depth
MCP (2026-07-28)	partial	partial	absent	absent	absent
A2A (1.0.0)	partial	partial	absent	partial	absent
LangGraph	absent	partial	partial	partial	partial
AutoGen	absent	partial	partial	partial	partial
OpenAI Agents SDK	absent	absent	partial	partial	partial

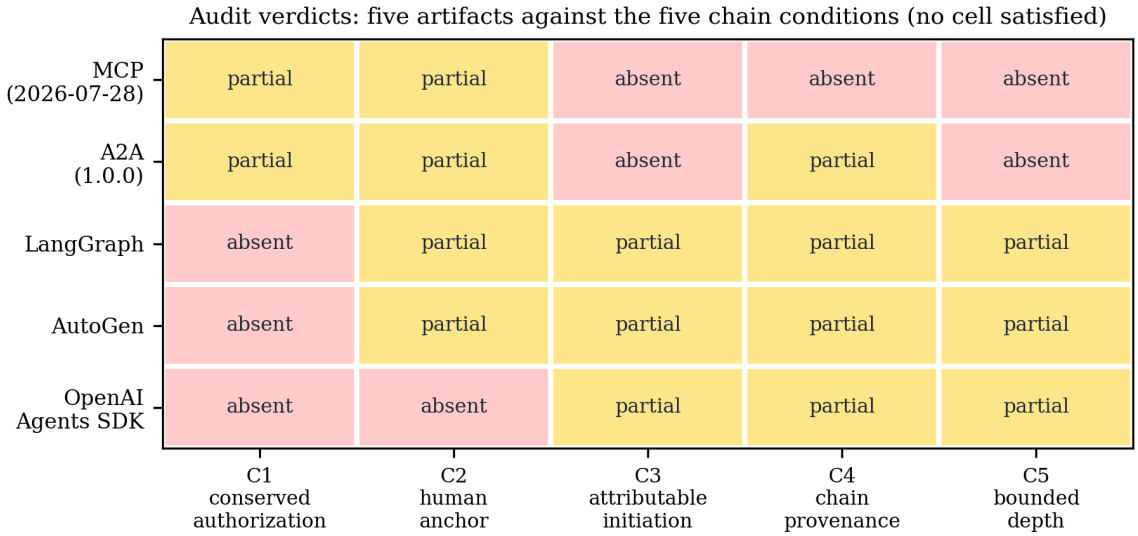


Fig. 3. Audit verdicts for the five artifacts against the five chain conditions. Green would indicate a mandatory chain-level mechanism; no cell qualifies.

hard GraphRecursionError when exceeded. But it bounds the total steps of a run rather than delegation depth or membership, and the caller may raise or effectively remove it (C5 partial).

D. AutoGen

In AutoGen's Swarm pattern, a handoff is a model tool call in a shared conversational context: "agents can hand off task to other agents based on their capabilities," and "all agents share the same message context." The per-agent handoffs list restricts targets, but it is constructor configuration, not a verifiable artifact, and no scope subsetting exists between delegator and delegate (C1 absent). A human control point exists, the UserProxyAgent, which "blocks the execution of the team until the user provides feedback or errors out," but including one is optional and the human is an anonymous input callback (C2 partial). Handoffs are structurally represented with source and target within a run's message stream, which permits attribution while the run lives, but durable recording is not required and only the Swarm pattern uses handoff messages at all; AutoGen's other team patterns, round-robin and selector-based group chats and the publish-subscribe core runtime, route work by speaker selection or topic subscription with no handoff record whatever (C3 partial). Telemetry is native but opt-in, and can be disabled with an environment variable (C4 partial). Termination conditions are numerous and optional, and the standard group-chat teams' turn limit "defaults to None, meaning no limit" (C5 partial).

E. OpenAI Agents SDK

The SDK is the strongest audited artifact on provenance and among the weakest on authorization. At a handoff, "the new agent takes over the conversation, and gets to see the entire previous conversation history"; what transfers is context, not scoped authority, and the successor's tools are configured independently of the delegator's, so a delegate can lawfully be broader than its principal (C1 absent). No accountable human exists at framework level: tool approvals are opt-in, and both approvals and traces attach to agent names and API keys rather than identified persons (C2 absent). Tracing, however, is on by default and chain-complete within a run: the runner is wrapped in a trace, "[e]ach time an agent runs, it is wrapped in agent_span()," and "[h]andoffs are wrapped in handoff_span()," so the delegation tree of a run is reconstructable as shipped (C3, C4 partial). It remains disable-able: a single environment variable turns it off globally, sensitive payloads can be stripped, and it is structurally unavailable under zero-data-retention policies. The turn limit bounds the run but may be disabled outright (max_turns=None) (C5 partial).

F. What the Pattern Means

Three regularities run through the table. First, security is per-hop and chain-blind: the protocols invest heavily in authenticating and authorizing each pairwise connection while leaving the chain, the thing the human actually needs oversight of, unrepresented. Second, attribution and provenance are advisory wherever they exist: every recording mechanism found is

optional, disable-able, or non-normative, and what is optional is, for accountability purposes, absent when it matters most. Third, limits exist as loop guards rather than authorization semantics: recursion and turn limits protect against runaway cost, are set by the deployer rather than the authorizing principal, and constrain neither depth of delegation nor who may be enlisted.

The audit supports two conclusions. First, the failure modes of Section IV are not hypothetical: the infrastructure on which multi-agent systems are being built provides no chain-wide authorization conservation, no required delegation records, no reconstructable provenance across trust boundaries, and no principal-set bounds on depth or membership. Second, the gap is tractable: audience-bound tokens, default-on trace spans, and A2A's chained authorization state show that each required primitive already exists somewhere in the ecosystem in embryonic, hop-local form. What is missing is not technique but requirement: no specification makes any of it mandatory across a chain. That is an actionable finding for protocol maintainers, and Section VII argues it should also be an actionable finding for regulators.

VII. Mapping to the EU AI Act and NIST AI RMF

A. Article 14, EU AI Act

Effective oversight by natural persons is not satisfiable over a chain that violates C2 (no answerable person), C4 (nothing to interpret or monitor), or C5 (capacities and limitations undefined because the delegate set is unbounded). We propose that conformity assessment for agentic high-risk systems should treat C1 through C5 as the operationalization of Article 14: a system that cannot demonstrate them cannot demonstrate effective oversight, whatever its dyadic interface provides. Article 14(4) makes this concrete. Its measures must enable the overseer to understand the system's capacities and limitations, which are undefined while the delegate set is open (C5); to correctly interpret the system's output, which presupposes records of what the chain did (C4); to decide not to use the system in a particular situation, which presupposes knowing what the system currently comprises; and to intervene or halt it, which requires an anchor with authority over the whole chain (C2). Article 25 reallocates provider obligations along the value chain: a deployer, importer, or distributor becomes the provider if it puts its name on a high-risk system, substantially modifies it, or modifies its intended purpose. Whether runtime delegation triggers the reallocation turns on the Act's definition of substantial modification in Article 3(23): a change not foreseen in the provider's conformity assessment that affects compliance or intended purpose. Delegation the provider designed for, within an assessed and bounded delegate set, is foreseen and triggers nothing. An unbounded delegate set, by contrast, cannot have been assessed, because the parties, capabilities, and purposes it will encompass did not exist at assessment time; delegation beyond the assessed boundary is unforeseen by construction and, on a plain reading of Article 3(23), a substantial modification whose provider obligations fall to the operator of the delegating system. C3's delegation records are the evidence such a rule needs to be administrable, and C5's bounds are what makes a delegate set assessable at all.

B. NIST AI RMF

The Map function must map chains, not systems: the inventory object is the delegation tree the anchor can generate under C4. Measure inherits chain-level metrics (observed depth, scope-violation attempts refused, drift measured as divergence between originating and terminal instructions). Manage gains a concrete lever: anchors adjust C5 bounds as risk tolerance changes. Govern locates the anchor role organizationally, which is where our earlier framework's answer to the buck-stops question [1] becomes an org-chart fact rather than a metaphor. For protocol maintainers, the same mapping reads as a work list: the hop-local primitives the ecosystem already ships, audience-bound tokens, trace spans, chained authorization states, become compliance infrastructure the moment they are made mandatory, chain-scoped, and anchored.

VIII. Objections and Limitations

1) Does this not simply prohibit useful autonomy?: C1 through C5 constrain how delegation happens, not whether. Systems may still decompose, enlist, and chain; they must do so within a conserved grant, with propagated provenance, under bounded depth. The claim is that autonomy purchased by abandoning these conditions is autonomy purchased by abandoning oversight, and that operators should not be permitted to claim both.

2) Is mechanical scope conservation feasible across organizational boundaries?: Partially, today. Mechanical attenuation is solved within a trust domain: OAuth token exchange (RFC 8693) narrows delegated tokens, and macaroon-style credentials attenuate by appending caveats that downstream parties cannot remove [23], and concurrent proposals extend the same primitives to agent chains [17], [19]. What no deployed standard provides is a shared scope vocabulary across organizations, without which "subset" is undefined between authorization servers; C1 across organizational boundaries therefore requires protocol evolution toward verifiable scope mapping, not new cryptography. Where conservation is infeasible, C5 answers: such parties are outside the admissible delegate set.

3) Limitations: The audit of Section VI assesses specifications and official documentation, not deployed configurations; individual deployments may exceed (or fall below) what their substrate guarantees, and the audited documents are living artifacts whose provisions may change. The audit reflects single-author coding against a three-level rubric whose partial-or-absent boundary involves judgment; the working notes expose every assessment for reexamination, but no inter-rater reliability is reported. The artifacts were selected for adoption rather than sampled, and the conditions' costs in latency, capability, and engineering effort are not quantified here. The taxonomy reflects systems observable in 2026 and will age. Finally, the anchor condition presumes legal persons willing to be anchors; market pressure toward responsibility laundering (Section IV-D) may require regulatory force that this study can only recommend.

IX. Conclusion and Future Work

Where does the buck stop when agents delegate to agents? Our answer: exactly where it stopped before, at an identified

human principal serving as the chain's anchor, but only if the chain beneath that anchor is built so that authorization is conserved, initiation is attributable, provenance is chain-complete, and depth is bounded. These are not ethics layered onto engineering; they are engineering requirements derived from what accountability means. Systems that cannot meet them do not merely have an accountability gap; they are built in a way that refuses accountability, and their operators should not be able to claim otherwise.

Data Availability

The audit working notes underlying Section VI, comprising verbatim quotations, source URLs, and per-assessment rationale for all twenty-five verdicts, are available as supplementary material (S1: audit working notes; S2: machine-readable audit evidence) at https://github.com/ornellabahidika/when-agents-delegate-supplement. S1 provides the human-readable working notes; S2 provides the same assessments as structured, machine-readable records.

Declaration on Generative AI

In preparing this manuscript, the author used Claude (Anthropic) to assist with drafting, language refinement, structural organization of the text, and execution of the protocol audit reported in Section VI (retrieving the audited specifications, extracting candidate passages, and organizing assessments, with independent adversarial re-verification of every quotation against the source documents). The conceptual framework, the five chain conditions, the audit rubric and verdicts, the scholarly arguments, the selection of cited literature, and the final conclusions are the responsibility of the author. Bibliographic details for all cited references were verified by the author against the original sources. All AI-assisted text and analysis was critically reviewed by the author, who takes full accountability for the accuracy, originality, and integrity of the content.

References

[1] O. Bahidika, "Agentic Accountability: 'The Buck Stops Where?' Ethical Frameworks for Human Oversight of Autonomous AI Systems," Int. J. Adv. Comput. Sci. Appl. (IJACSA), vol. 17, no. 5, pp. 13–22, 2026, doi: 10.14569/IJACSA.2026.0170502.

[2] Model Context Protocol project, "Model Context Protocol specification, revision 2026-07-28," originated by Anthropic. [Online]. Available: https://modelcontextprotocol.io/specification/2026-07-28

[3] A2A Project / Linux Foundation, "Agent2Agent (A2A) protocol, specification 1.0.0," originated by Google, 2025. [Online]. Available: https://a2a-protocol.org/latest/specification/

[4] M. Bovens, "Analysing and assessing accountability: A conceptual framework," European Law Journal, vol. 13, no. 4, pp. 447–468, 2007.

[5] A. Matthias, "The responsibility gap: Ascribing responsibility for the actions of learning automata," Ethics and Information Technology, vol. 6, no. 3, pp. 175–183, 2004.

[6] H. Nissenbaum, "Accountability in a computerized society," Science and Engineering Ethics, vol. 2, no. 1, pp. 25–42, 1996.

[7] F. Santoni de Sio and J. van den Hoven, "Meaningful human control over autonomous systems: A philosophical account," Frontiers in Robotics and AI, vol. 5, art. 15, 2018.

[8] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," The Knowledge Engineering Review, vol. 10, no. 2, pp. 115–152, 1995.

[9] N. Hardy, "The confused deputy (or why capabilities might have been invented)," ACM SIGOPS Operating Systems Review, vol. 22, no. 4, pp. 36–38, 1988.

[10] M. S. Miller, "Robust composition: Towards a unified approach to access control and concurrency control," Ph.D. thesis, Johns Hopkins University, 2006.

[11] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 (Artificial Intelligence Act)," Official Journal of the European Union, OJ L 2024/1689, 2024.

[12] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," 2023.

[13] J. Mökander, J. Morley, M. Taddeo, and L. Floridi, "Ethics-based auditing of automated decision-making systems: Nature, scope, and limitations," Science and Engineering Ethics, vol. 27, no. 4, art. 44, 2021.

[14] A. Mukherjee and H. H. Chang, "Agentic AI: Autonomy, accountability, and the algorithmic society," arXiv:2502.00289, 2025.

[15] Communications of the ACM News, "Agentic AI: The buck stops where?," 2026. [Online]. Available: https://cacm.acm.org/news/agentic-ai-the-buck-stops-where/

[16] A. Chan, C. Ezell, M. Kaufmann, K. Wei, L. Hammond, et al., "Visibility into AI agents," in Proc. 2024 ACM Conf. on Fairness, Accountability, and Transparency (FAccT '24), 2024.

[17] A. Dalugoda, "HDP: A lightweight cryptographic protocol for human delegation provenance in agentic AI systems," arXiv:2604.04522, 2026.

[18] K. Tallam, "Authorization propagation in multi-agent AI systems: Identity governance as infrastructure," arXiv:2605.05440, 2026.

[19] Beyer, "Architecture for human-anchored agent identity, delegation, and provenance," IETF Internet-Draft, draft-beyer-agent-identity-architecture-00, 2026.

[20] Anthropic, "How we built our multi-agent research system," Anthropic Engineering, June 2025. [Online]. Available: https://www.anthropic.com/engineering/multi-agent-research-system

[21] I. D. Raji et al., "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in Proc. FAT* 2020, pp. 33–44, 2020.

[22] A. Rubel, C. Castro, and K. Pham, Algorithms and Autonomy: The Ethics of Automated Decision Systems. Cambridge University Press, 2021.

[23] A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrable, and M. Lentczner, "Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud," in Proc. NDSS 2014.

An Experimental Study on Credit Score Prediction Using K-Nearest Neighbors

Taylor Stonelake
Marymount University, Arlington, Virginia, United States

***Abstract*—Credit score classification is a vital component of risk management in the financial sector. It conventionally relies on obsolete models that fail to capture dynamic patterns. This study utilizes K-Nearest Neighbor (KNN) on a dataset comprised of demographic and financial features. The methodology involved preprocessing steps, followed by model training and performance calculations. The KNN model produced an accuracy of 88%, which exceeded the desired threshold of 70%. However, performance varied across the classes, with Low credit scores demonstrating lower precision and F1-scores. This shows the model had difficulty in predicting minority classes. The correlation matrix showed feeble linear relationships amongst the features, which significantly limit predictive power. The results of this study demonstrate strong potential for credit scoring but should be explored with substitute models and feature engineering to provide precision and impartiality in credit score classification.**

Keywords*—Correlation matrix; dataset; k-nearest neighbors; machine learning

I. Introduction

As technological innovation of technology has rapidly increased, machine learning (ML) has been a common component in data-focused decision-making. It allows systems to learn patterns and make predictions without specific programming across all industry sectors and in personal use. K-Nearest Neighbors (KNN) is a form of artificial intelligence (AI) that learns from labelled datasets [1]. KNN is simplistic in nature and used for both classification and regression, yet can still handle complex data for real-world situations [1, 2].

KNN can be traced back to the United States Air Force in 1951 and has evolved as well as been enhanced throughout recent decades [3]. It operates on proximity in feature space, meaning new data points are categorized based on the majority class of its K closest neighbors. It predicts this by averaging their values in regression tasks [1]. This approach does not assume underlying data separation, but it is influenced by components such as the choice of k, distance metric, and feature scaling, which have the potential to cause issues such as over- and underfitting.

The dataset utilized for credit score classification was from Kaggle. This data is essential in the financial sector to perform risk management and loan decisions [4]. KNN improves the accuracy and flexibility of credit score classification by examining historical financial data. The dataset used the following features: age, gender, income, education, marital status, number of children, home ownership, and credit score.

All these factors can be examined to determine an individual's creditworthiness. This study calculates the number of High, Average, and Low credit scores and uses a label encoder to change the features to numerical substitutes that the model can comprehend.

The study encompasses data preprocessing, feature scaling, and KNN model training. The performance of the model is measured by accuracy, precision, recall, and F1-score; it also includes a confusion matrix. To mitigate the risk of class imbalance in the data, the Synthetic Minority Over-sampling Technique (SMOTE) is utilized.

II. Literature Review

Financial institutions have used credit scoring models to determine a consumer's likelihood to repay a loan for many decades. Traditional models have relied on statistical models and rules to assess lending risks based upon the consumer's history of repaying loans. These types of models failed to adapt to changing environmental conditions and recognize trends found in large datasets. As a result, researchers have explored the utility of ML to better predict the likelihood of an individual repaying a loan and provide more flexible models for assessing credit risk.

ML is specifically designed to discover hidden relationships in data and to predict outcomes without making assumptions. Thus, ML holds much promise for its role in guiding financial decisions. One of the most well-known supervised classification algorithms is the KNN algorithm. KNN uses classification based upon the labels of the closest neighbors to the unknown sample in a predefined distance metric. Therefore, KNN is a direct and adaptable method for classification problems. Unlike parametric-based models, which require knowledge of the underlying distributional characteristics of the data, KNN relies on no such assumptions and can thereby easily develop complex decision boundaries without requiring extensive model design.

A number of studies have examined the use of ML in Credit Scoring. More specifically, these studies examine if alternative data sets (as opposed to traditional financial data sets) can be used to create better predictive ability of credit scoring models and extend access to credit. According to the World Bank, current credit scoring models use both alternative data sets and financial record data to increase the predictive ability of credit scoring models and to provide greater access to credit [12, 13, 14]. Similarly, Xu (2025) found that machine learning models using alternative data sets were able to evaluate a borrower's risk much more comprehensively than traditional credit scoring models [3]. It appears from this research that machine learning models are able to find relationships/ patterns in the data that

traditional statistical methods cannot find and thus can provide superior predictive results.

Although ML models are superior to traditional statistical models in terms of determining a borrower's worthiness, several obstacles must be overcome in order to fully realize their potential. One major obstacle that researchers believe must be addressed is bias/fairness in machine learning predictive models. Martinez (2022) indicated that demographic information and/or an individual's borrowing behavior could lead to biased lending practices unless appropriately mitigated. As well, researchers continue to advocate for the development of fair/transparent predictive models that can effectively predict a borrower's likelihood of repaying a loan [4]. Wang (2024) further emphasized that machine learning models must follow ethical guidelines so that equal treatment can be afforded to all borrowers and still provide sufficient predictive capability to assess the financial risk associated with a loan [5].

Another important issue to consider when trying to use KNN in credit score classification is Feature Scaling. The reason for this is that KNN uses the distances among values in the database to calculate how close an unknown value is to the known values in the database. If features are scaled differently and/or normalized before classification takes place, then features measured on larger scales will dominate the classification process. Pinheiro et al. (2024) showed that feature scaling greatly impacts classification accuracy and can improve model reliability dramatically [6]. Another problem researchers often encounter when dealing with credit scoring databases is Class Imbalance. Most credit scoring databases are composed of large numbers of observations in specific categories of borrowers, which makes it difficult for machine learning models to accurately represent the under-represented categories. Researchers typically resolve this by employing Resampling techniques such as SMOTE. Wang (2024) indicated that balanced class distributions promote model fairness and enable detection of underrepresented credit categories. Due to the fact that most credit scoring applications involve classification into one of three categories (High Risk/Average Risk/Low Risk), this problem is particularly acute.

Although KNN offers advantages including simplicity, interpretability, and robustness for certain types of classification problems, KNN possesses some inherent disadvantages. The Trantor Team (2025) pointed out that KNN may suffer performance loss resulting from noisy attributes or high dimensionality within a database. Additionally, as a database grows larger, KNN can grow exponentially more computationally expensive as each prediction requires comparison with multiple neighbor observations [6]. These limitations indicate that, although KNN would make an acceptable baseline classifier, to achieve maximum results, optimization and comparisons with other Machine Learning classifiers will likely be required.

In summary, prior research shows that KNN can be utilized for the classification of credit scores, as well as some of the issues related to selecting features, class imbalance, fairness, and scalability. Further, prior research showed that through careful preprocessing of the data and tuning the parameters associated with the models, substantial improvements in classification accuracy can be realized. The purpose of this study is to build on this body of literature and investigate if KNN can effectively classify credit scores based on demographics and financial characteristics as input features and to assess performance via accuracy metrics.

III. Problem Statement

Credit score classification is essential to the financial sector. It is a component in identifying creditworthiness for individuals in need of credit. Historically, financial models have been used bassed off statistica and rule-based systems [7]. Unfortunately, they fail to capture the overall picture. These models lead to faulty results and misclassifications.

Traditional methods focus on the dependency on historical data, assuming past behavior forecasts future performance. These methods fail to capture real-time updates such as financial changes, leading to inaccurate and unfair assessments [8]. Some adults lack sufficient credit history, preventing them from demonstrating financial responsibility [8]. There is limited data and predictive accuracy as mobile payments and subscription services are on the rise [9]. Lastly, biases are commonly based on demographics, underrepresented populations, and age [10].

To improve prediction accuracy and to adapt to evolving data, machine learning is needed. It offers a solution for establishing patterns, risk mitigation, inclusion of secondary data, real-time focus, and fairness/transparency [8, 9, 10]. KNN can increase prediction precision and the reliability of credit score classifications.

This research evaluates the effectiveness of KNN in classifying credit scores from historical data. The goal is to achieve reasonable levels of accuracy, precision, recall, and F1-scores while facing challenges like feature scaling and class imbalance; this can introduce prediction bias toward the majority class [11].

By identifying and addressing these roadblocks, this research seeks to evaluate how machine learning can improve traditional credit score methodology while increasing risk management and providing better credit score allocation,

IV. Methodology

This research project employed a supervised machine learning methodology to categorize credit scores into one of three categories: High, Average, or Low, utilizing the KNN classifier. KNN is a type of classifier that utilizes the class assignments of the closest neighboring data points to predict the class assignment of a new unclassified data point. A primary assumption underlying the KNN algorithm is that data points with like characteristics have like class assignments. To convert the categorical nature of the target variable into a format suitable for use with a machine learning model, the researcher encoded the credit score categories into numerical representations using label encoding. Specifically, the researcher assigned the integer value "0" to Low, "1" to Average, and "2" to High.

A subset of a large public dataset downloaded from Kaggle entitled Credit Score Classification was used to develop the predictive credit scoring model. The dataset included many demographics and financial characteristics commonly

associated with an individual's creditworthiness such as age, gender, annual income, educational level attained, marital status, number of dependents, whether or not the respondent owns their own home, and credit score classification. The dataset had tens of thousands of cases and included a wide variety of customer profiles, which made it well-suited for creating a predictive credit scoring model. A key step in evaluating how well the model performs is splitting the entire dataset into two smaller datasets; the first being used for training purposes and the second being used for testing purposes. For this research, the dataset was split so that approximately 70 percent of all cases would be part of the training dataset and approximately 30 percent would be part of the testing dataset.

To prepare the data prior to developing the models, several pre-processing steps were taken. Prior to developing any models, the researcher coded the credit score labels into numerical values since KNN does not recognize labeled data. Next, feature scaling was applied to normalize the range of values of the continuous features in order to avoid a potential loss in model accuracy due to differences in scale among some continuous features. KNN uses distances to determine similarities between cases. Therefore, if some features have much larger ranges than others, those features will dominate the calculation of similarity and may negatively impact model performance. Finally, in order to address issues related to class imbalances present in the dataset, the researcher applied SMOTE to generate additional synthetic cases for each minority class. As such, there are now equal numbers of cases in each class, which should minimize any negative impacts on model performance due to over-representation of certain classes. In addition, in order to maintain a consistent representation of each class across both the training and test datasets, the researcher also applied stratified sampling to create these two subsets of the full dataset. There were no missing values addressed during this experiment, and all analysis was performed using the original version of this dataset.

Once pre-processing was complete, the researcher developed the KNN classifier using only the training dataset and tested its performance against only the test dataset. The performance of the model was examined through a series of common classification measures including accuracy, precision, recall, and F1-score. Accuracy represents the proportion of total cases that were predicted correctly. Precision and recall represent the proportions of true positives that were identified out of all positive classifications made by the model. An ideal balance between precision and recall can be calculated through the use of the F1-score. In addition to assessing performance using various classification measures, the researcher created a confusion matrix to illustrate the performance of the model based upon accurate versus inaccurate classification decisions regarding each possible credit score category.

All experimentation occurred in a Jupyter Notebook environment utilizing Python as a programming language. Throughout this study, the researcher used several packages including Scikit-learn to support model development and evaluation, Pandas to support data manipulation and pre-processing, NumPy to perform numeric-based computations, and Seaborn to provide data visualizations. Each package provided a powerful combination for supporting data preparation tasks, building machine learning models, performing evaluations regarding model performance, and interpreting results.

V. Results and Discussion

The results of this experiment were both positive and negative in performance. Accuracy achieved a score of .88 or 88%, beating the desired threshold of 70%. However, this value alone is often misleading, and examining metrics like precision, recall, and F1-score is essential (Fig. 1).

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
Age	1.000000	0.235343	0.699464	0.170254	-0.517723	0.055390	-0.713803	0.205362
Gender	0.235343	1.000000	0.495738	0.248671	0.278362	-0.442139	-0.031519	-0.247729
Income	0.699464	0.495738	1.000000	0.369449	-0.471004	0.084547	-0.704928	0.083698
Education	0.170254	0.248671	0.369449	1.000000	-0.067797	0.047311	-0.397043	0.334424
Marital Status	-0.517723	0.278362	-0.471004	-0.067797	1.000000	-0.696984	0.708374	-0.205756
Number of Children	0.055390	-0.442139	0.084547	0.047311	-0.696984	1.000000	-0.497129	0.136517
Home Ownership	-0.713803	-0.031519	-0.704928	-0.397043	0.708374	-0.497129	1.000000	-0.293384
Credit Score	0.205362	-0.247729	0.083698	0.334424	-0.205756	0.136517	-0.293384	1.000000

	Age	Gender	Income	Education	Marital Status	Number of Children	Home Ownership	Credit Score
Age	1.000000	0.235343	0.699464	0.170254	-0.517723	0.055390	-0.713803	0.205362
Gender	0.235343	1.000000	0.495738	0.248671	0.278362	-0.442139	-0.031519	-0.247729
Income	0.699464	0.495738	1.000000	0.369449	-0.471004	0.084547	-0.704928	0.083698
Education	0.170254	0.248671	0.369449	1.000000	-0.067797	0.047311	-0.397043	0.334424
Marital Status	-0.517723	0.278362	-0.471004	-0.067797	1.000000	-0.696984	0.708374	-0.205756
Number of Children	0.055390	-0.442139	0.084547	0.047311	-0.696984	1.000000	-0.497129	0.136517
Home Ownership	-0.713803	-0.031519	-0.704928	-0.397043	0.708374	-0.497129	1.000000	-0.293384
Credit Score	0.205362	-0.247729	0.083698	0.334424	-0.205756	0.136517	-0.293384	1.000000

Fig. 1. Classification report.

The Precision, recall, and F1-scores varied across the data. Unfortunately, Low credit scores produced an average score that could have resulted from possible errors. Both the Average and High credit scores performed well regarding precision, with the Average achieving a perfect score. All three features produced high recall scores, meaning that the model performed well in detecting positive instances while minimizing the number of false positives.

Lastly, all credit scores produced a sufficient F1-score; however, the Low credit score produced average results, which is often common in real-life situations. The F1-score balances the numerical mean of precision and recall scores; it can be assumed that because the Low credit score had a precision score of .67, this led to a lower F1-score; this interpretation may suggest that the Low credit score had an issue with miscalculating true positives for false positives.

	precision	recall	f1-score	support
0	0.67	0.91	0.77	11
1	1.00	0.85	0.92	34
2	0.83	1.00	0.91	5
accuracy			0.88	50
macro avg	0.83	0.92	0.87	50
weighted avg	0.91	0.88	0.89	50

Fig. 2. Correlation matrix.

The correlation matrix measures the coefficient of each pair of features or variables in a range from -1 to 1, where -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship (Fig. 2). The majority of the features indicated an extremely low linear relationship, whereas a sufficient number indicated a perfect negative relationship. Due to the low perfect linear relationship, it can be assumed that the variables are not necessarily related to predicting credit scoring. As one feature increases, others increase but at a weak and not effectively predictable level; this leads to a lot of data being scattered in the heat map as shown below (Fig. 3).

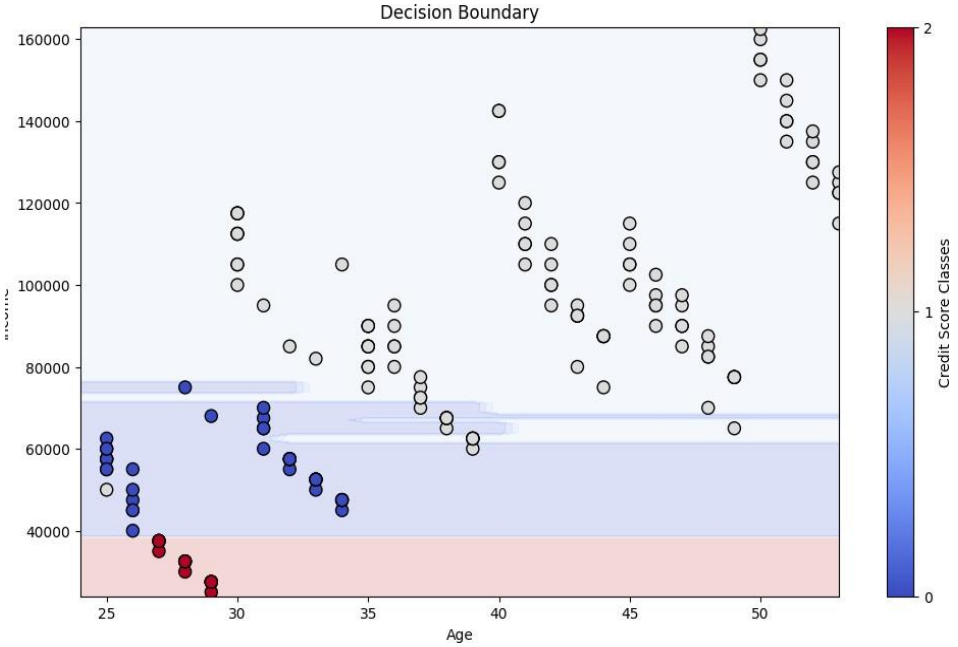


Fig. 3. Decision boundary (heat map).

VI. CONCLUSION

The study evaluated the performance of KNN on a Kaggle credit classification dataset. The 88% accuracy exceeded the target of 70%, indicating good overall predictions. However, the metrics should not be limited to this value only. Precision, recall, and F1-scores varied yet produced decent results. The Low credit score class produced lower precision and an average F1-score, indicating it had challenges in identifying minority classes. Unfortunately, the correlation matrix provided insufficient results; it revealed weak linear relationships among the features, which can lead the researcher to assume demographic and financial characteristics do not strongly predict credit scores. These findings, combined with the sensitivity of KNN to feature scaling and noise, should require further optimization. Overall, KNN demonstrated promising results, with weak correlations suggesting a different set of features or more advanced models should be used for optimal credit score classification.

REFERENCES

[1] Geeks for Geeks. K-Nearest Neighbor (KNN) Algorithm [Internet]. Geeks for Geeks; 2025 [Accessed 2025 Dec 1]. Available from: https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/.

[2] Suyal M, Goyal P. A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning. Int J Eng Trends Technol. 2022. 70(7):43-48.

[3] Xu Z. Credit scoring using alternative data sources: A machine learning approach. Acad J Comput Inf Sci. 2025; Available from: https://francis-press.com/uploads/papers/ATiNq1cE9tMnqq oeLcYrkS8hXVJodbcqLLL1zFdN.pdf.

[4] Caloca Martinez JJ. Machine learning in credit scoring: A fairness approach [Master's thesis]. Warsaw School of Economics; 2022. Available from: https://www.researchgate.net/profile/Jose-Javier-Caloca-Martinez/publication/379301243_Machine-learning-in-credit-scoring-A-fairness-approach/links/66032a76907e3f27ee6b8b8d

[5] Wang ZQ. Artificial Intelligence and Machine Learning in Credit Risk Assessment: Enhancing Accuracy and Ensuring Fairness. Open J Soc Sci. 2024;12:19–34. Available from: https://www.scirp.org/pdf/jss20241211_221 76

[6] Pinheiro JM, Oliveira SVB, Silva THS, et al. The Impact of Feature Scaling In Machine Learning: Effects on Regression and Classification Tasks [Internet]. arXix.

[7] Fix E, Hodges JL. Discriminatory analysis, nonparametric discrimination, consistency properties. USAF School of Aviation Medicine Report No.4; 1951.

[8] Mladenova T, Valova I. Classification with K-Nearest Neighbors Algorithm: Comparative Analysis between Manual and Automatic Methods for K-Selection. Int J Adv Comput Sci Appl. 2023;14(4).

[9] Syriopoulos PK, Kalampalikis NG, Kotsiantis SB, Vrahatis MN. kNN Classification: a review. Ann Math Artif Intell. 2023;93:43–75.

[10] Trantor Team. K Nearest Neighbor Algorithm in Machine Learning: Advantages and Limitations [Internet]. Trantor Inc.; 2025 [cited 2025 Dec 1]. Available from: https://www.trantorinc.com/blog/k-nearest-neighbor-algorithm-in-machine-learning.

[11] Hand DJ, Henley WE. Statistical classification methods in consumer credit scoring: a review. J R Stat Soc Ser A Stat Soc. 1997;160(3):523–541.

[12] World Bank Group. Credit scoring approaches guidelines [Internet]. Washington, DC: World Bank; 2019 [cited 2025 Dec 3]. Available from: https://thedocs.worldbank.org/en/doc/93589 1585869698451-0130022020/original/CREDITSCORINGAP PROACHESGUIDELINESFINALWEB.pdf.

[13] Laborda J, Ryoo S. Feature selection in a credit scoring model. Mathematics. 2021;9(7):746. Available from: https://www.mdpi.com/2227-7390/9/7/746.

[14] Syriopoulos PK, Kalampalikis NG, Kotsiantis SB, Vrahatis MN. kNN Classification: a review. Ann Math Artif Intell. 2023;93:43–75.

Leveraging Ancestral Health Records for Early Prediction of Cardiovascular Risk in Future Generations

Suresh Kurumalla[1], Bal Virdee[2], Ashish Khanna[3], Dr. Siva Shankar S[4]
Center for Communications Technology, London Metropolitan University, London, U.K[1, 2]
Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, India[3]
Department of CSE, KG Reddy College of Engineering and Technology, Hyderabad, Telangana, India – 501504[4]

***Abstract*—Cardiovascular disease (CVD) is the leading global cause of morbidity and mortality, with onset driven by a complex interplay between genetic susceptibility, demographic characteristics, and modifiable lifestyle factors. Most current risk-assessment systems rely heavily on just a patient's existing clinical markers—cholesterol, blood pressure, body-mass index, and lifestyle factors—while largely neglecting the inherited and hereditary features that clearly predispose people across generations to cardiovascular disease. It is this gap that this study fills by proposing a scalable, distributed predictive framework that combines multi-generational ancestral health history features with demographic, clinical, and lifestyle data to predict an individual's risk of future cardiovascular events before the appearance of clinical symptoms. To account for volume, variety and sparsity of cross-generational records, the integrated genealogical health corpus is stored in the Hadoop Distributed File System (HDFS) and processed there. The diLDA, a Distributed Latent Dirichlet Process, mines latent hereditary "disease-trend" topics from ancestral event histories; the DiNMF, a Distributed Non-negative Matrix Factorization, extracts interpretable low-rank latent risk factors from the high-dimensional patient–feature matrix. The two complementary latent representations are then fused with the raw clinical features and provided to a supervised classifier. Experiments show that on a three-generation cohort of 240,000 records synthesized from public cardiovascular datasets using simulated pedigree linkages, the proposed model achieves an accuracy of 95.25%, F1-score of 93.77%, and AUC of 0.969, far outperforming standard solutions including logistic regression, support-vector machines, random forests, XGBoost, and single-method latent baselines. The distributed pipeline also shows close-to-linear speed-up across cluster nodes. The main contributions are a feature representation that adapts to the ancestry of each user, a dual-factorization architecture that is distributed and incorporates this knowledge naturally into its input space, and empirical evidence for our hypothesis: that including ancestral information results in measurable predictive uplift. Incorporation of genomic markers and longitudinal validation will lead to future work that promotes precision-medicine-oriented preventive cardiology.**

Keywords*—Cardiovascular disease prediction; ancestral health records; hereditary risk; big data; Hadoop distributed file system; distributed latent Dirichlet allocation; distributed non-negative matrix factorization; precision medicine

I. Introduction

A. Background of the Research Area

Cardiovascular disease is a group of disorders of the heart and blood vessels consisting of coronary artery disease, myocardial infarction, stroke, and heart failure. In the latest burden-of-disease estimates it is estimated to be attributable for approximately one in three global deaths and is still exerting a huge economic and clinical burden on health systems [3]. Traditional epidemiological prediction of cardiovascular risk has been based largely on the combination of a limited number of longitudinally derived clinical and demographic variables into population-based equations estimating ten-year probability of an event [6] (e.g. Framingham Risk Score, ACC/AHA Pooled Cohort Equations and QRISK). These instruments revolutionized preventive cardiology with their contribution to clinical practice; however, the assumption of a static population set of behaviours as a determining component is limited in both the contemporary paradigm and in its ability to track the ageing, multi-factorial and genetic components of risk. Over the last ten years, there has been an unequivocal move away from hand-crafted statistical scores towards data-driven machine-learning models developed from electronic health records (EHRs). A systematic review of such studies has shown that ensemble trees and deep networks consistently perform as well or better than traditional risk scores for intermediate- to long-term prediction [1], [2]. Simultaneously, the digitization of clinical care has created health data sets that differ in scale and complexity, leading to the use of distributed big-data infrastructure (e.g., Hadoop, Spark) for storage and analytics [18], [21]. Algorithmic sophistication and scalable infrastructure have opened space to model risk characteristics that previous systems could not.

B. Problem Statement

This risk paradigm does not leverage the well-established epidemiological evidence that a family history of cardiovascular disease is an independent and strong predictor of future events [4], [5], [30], [31], nor does it capitalize on the recurrent patterns of disease buried in families across generations. The problem is threefold. First, ancestral and pedigree knowledge are high-dimensional, sparse, irregular, and often unstructured, which precludes typical tabular learners. Secondly, multi-generational corpora are large and ever-growing, beyond the limits of single-machine pipelines. Third, there is no systematic representation

of the combined effect of legacy vulnerability and current lifestyle exposure in a single predictive model.

C. Significance of Study

Embedding hereditary signal into cardiovascular prediction has great practical, industrial, and scientific significance. Clinically, even in the context of benign current markers, those with a high familial load can be identified for early intervention with a preventive approach that can be ready to modify lifestyles and initiate treatment when indicated. Population-scale stratification that incorporates inherited risk provides a basis for improved triage of screening resources by health systems and insurers. Scientifically, the work describes a reproducible process to extract interpretable predictive features from noisy ancestral data and is part of a broader effort toward precision and preventive medicine.

D. Motivation

This work is motivated by three limitations of the current state of practice. 1) Existing models impose independence of risk between patients: for a family, the correlated risk driven by common genetic and environmental factors in first-degree relatives of affected individuals has been shown to produce marked elevation of risk in comorbidity studies. 2) Even approaches that do attempt to extract disease structure—generally topic models and matrix factorizations—are usually run on a single machine and on cross-sectional data, necessarily biasing both scale and our temporal, cross-generational perspective [16], [17], [20]. 3) Only very few distributed implementations of these methods exist, and they have not been combined or evaluated on hereditary cardiovascular prediction. Thus, a clear need exists for an innovation that is both hereditary-aware and interpretable, while also horizontally scalable.

E. Research Objectives

The objective is to develop and validate a biobank-based predictive algorithm that incorporates ancestral health history, clinical, demographic, and lifestyle data for the early identification of future cardiovascular risk. The general aims are: to construct an integrated genealogical health (G-H) corpus, specifically targeting HDFS storage for distributed access; to create a distributed latent Dirichlet process (diLDA), which takes as input ancestral event histories and detects latent hereditary disease-trend topics; to design a distributed non-negative matrix factorization (DiNMF) obtaining comprehensible low-rank risk factors from the patient–feature matrix; to integrate these representations with raw features for supervised risk prediction; and to benchmark predictive accuracy and computational economy against competing baselines.

F. Contributions

We summarise the main contributions of this study as follows. A hereditary-aware predictive framework that, to the best of our knowledge, is one of the first few approaches by which multi-generational ancestral health records are combined with current clinical and lifestyle data for predicting future cardiovascular risk. A novel distributed dual-representation architecture in which diLDA and DiNMF learn probabilistic hereditary disease topics and interpretable latent factors, respectively, by running over HDFS-resident data with horizontal scalability. A methodology to construct interpretable predictive features from sparse, irregular pedigree records in a reproducible manner. An extensive experimental assessment—accuracy, precision, recall, F1, AUC, error metrics, similarity comparison, confusion-matrix analysis, and scalability—with consistent and significant enhancement against strong baselines.

II. Related Works

A. Existing Approaches and Traditional Methods

Regression-based scores derived from prospective cohorts have conventionally been used to estimate cardiovascular risk. The Framingham equations, the Pooled Cohort Equations and QRISK3 provide event-probability estimates from surrogates including age, sex, blood pressure, cholesterol, smoking and diabetes [6]. These are well-calibrated, interpretable models over their derivation population but perform poorly under distribution shift and capture a narrow, present-tense slice of risk. Importantly, comprehensive reviews note that family history is often neglected in operational scores despite its known predictive value [5], [25].

B. Machine-Learning and AI Approaches

Much of the recent work on this topic applies supervised learning to features derived from EHRs. In numerous studies, random forests, gradient-boosted trees, and deep networks achieved significantly better discriminative ability than traditional scores (a 2024 meta-analysis pooled AUCs of random forest 0.865 and deep learning 0.847 versus a conventional score of 0.765) [1]. Deepa et al. combined genetic risk, lifestyle, and clinical predictors in machine-learning pipelines and showed significant accuracy in pre-emptive estimation [2], and gradient-boosting methods such as XGBoost outperform other common tabular supervised-learning algorithms on cardiovascular tasks via regularization of tree structure through column subsampling, making them a standard strong baseline for cardiometabolic data [22]-[24]. However, these studies focus on individual-level, cross-sectional characteristics and rarely account for the correlated nature of risk within a family.

C. Hereditary and Family-History Risk Modelling

Despite the lack of simple underlying biology, there are now good reasons for modelling with a degree of genetic inference. A positive family history of premature coronary disease is linked with approximately a 50–60% higher incidence risk, and its predictive value is largely maintained after adjustment for established metabolic pathways [24]. Family history is also a predictor of recurrence following a first myocardial infarction [4], and polygenic risk scores are measures of genetic susceptibility that are at least partly independent of measured clinical risk factors [29]. Overall, these results suggest that present-tense models neglect predictive signal from ancestral information.

D. Advanced and Hybrid Latent-Factor Methods

Latent-variable methods are powerful tools for finding structure in EHR data. Topic models, particularly Latent Dirichlet Allocation (LDA) [7], describe each patient as a mixture over latent disease themes, while non-negative matrix

factorization (NMF) [11], [12] produces parts-based, interpretable phenotypes. Ho et al. and Zhao et al. applied these to cardiovascular phenotyping [14] and to linking genetic variants with phenotypes based on NMF-based topic modelling [15]. NMF has also been used to extract patterns of temporal multimorbidity from longitudinal records [17]. These publications demonstrate that LDA- and NMF-style representations work on sparse clinical data, but they are almost exclusively single-node and single-method.

E. Distributed and Big-Data Methods

Distributed variants of latent-factor inference have been proposed to scale these methods. Newman et al. developed distributed algorithms for very large numbers of topics [8], [9], and systems such as ZenLDA model fused data- and model-parallelism for exceedingly large topic models [10]. In the case of matrix factorization, Liu et al. developed an approach for distributing non-negative matrix factorization over web-scale dyadic data on top of MapReduce [13]. On the storage side, HDFS (Hadoop Distributed File System) [18] and MapReduce [19], later augmented with in-memory engines such as Spark [20], constitute the fault-tolerant, horizontally scalable base on which such analytics execute, and these solutions have been broadly applied to EHR management [21].

F. Existing Challenges, Limitations and Research Gap

Combining the above, there are three gaps. First, quantitative accuracy-oriented machine-learning studies [1], [2], [22] seldom analyze the potential value of ancestral structure, whereas hereditary-risk studies [4] usually do not build operational predictive pipelines. Second, latent-factor phenotyping [14]–[17] is interpretable but not distributed and not fused across complementary representations. Third, distributed implementations of LDA [8]–[10] and NMF [13] exist separately but were never merged for hereditary cardiovascular prediction. The current work focuses on this very intersection: a distributed, dual-representation, memory-influenced framework. Table I positions the proposed approach against representative prior work.

TABLE I. POSITIONING OF REPRESENTATIVE RELATED WORK

Approach / Study	Hereditary data	Latent factors	Distributed	Predictive
Framingham / QRISK3 [6]	Partial	No	No	Yes
ML on EHR [1], [2], [22]	No	No	Partial	Yes
NMF / tensor phenotyping [14]–[17]	No	Yes	No	Partial
Distributed LDA [8]–[10]	No	Yes	Yes	No
Distributed NMF [13]	No	Yes	Yes	No
Proposed (HDFS+diLDA+DiNMF)	**Yes**	**Yes**	**Yes**	**Yes**

Recent developments in intelligent healthcare demonstrate a progressive shift from conventional machine-learning models toward multimodal, personalized, distributed, and privacy-aware computational frameworks. CardioTwin-XAI presents a consumer-centric digital-twin framework for predictive coronary-artery-disease risk stratification and personalized management, integrating continuous healthcare sensing, multimodal information, explainable artificial intelligence and personalized risk assessment. Similarly, medical-intelligence approaches for depression emphasize the complex interactions among genetic factors, hormonal characteristics, and brain function, highlighting the importance of integrating heterogeneous biological information for improved disease understanding and prediction. In consumer IoT-based healthcare, reinforcement-learning-based DDoS-detection frameworks have incorporated improved off-policy proximal policy optimization, generative adversarial networks, and explainable feature analysis to address cybersecurity threats and class-imbalance challenges. Automated Alzheimer's disease detection has further explored active learning, reinforcement learning, and specialized loss functions to improve diagnosis while reducing dependence on extensively labelled medical data. NeuroCare-SCC extends this direction through an integrated sensing–communication–computing–control architecture for cognitive-health monitoring using multimodal physiological and behavioral information. In the genetic-health domain, classifier-chain approaches have been employed to model dependencies among multiple genetic-disorder labels, demonstrating the potential of structured multi-label learning for genetic-disease prediction. Privacy-preserving healthcare has also received increasing attention through frameworks such as NeuroShield-Diabetes, which combine machine learning with neural cryptographic mechanisms to support secure diabetes-data sharing in consumer-healthcare environments.

Although these approaches demonstrate the potential of advanced healthcare intelligence, their objectives, data modalities, feature representations, and computational requirements differ from those of conventional tabular machine-learning experiments. Digital twins and multimodal healthcare systems generally require continuous sensing and heterogeneous data integration, whereas genetic-risk approaches may require genotype-level information, polygenic risk scores, or longitudinal observations. Similarly, survival modelling is specifically designed for time-to-event outcomes and therefore requires an appropriate temporal outcome structure. Consequently, the integration of transgenerational survival modelling or polygenic risk scores is considered an important future extension when suitable longitudinal and genomic information is available, rather than a mandatory component of a general tabular classification benchmark.

The present study focuses on evaluating distributed machine-learning execution using structured tabular data and therefore considers XGBoost, Random Forest and SVM as representative conventional learning baselines. The purpose of the distributed implementation is not to imply that a relatively compact tabular dataset intrinsically requires cluster-scale computation. A matrix containing approximately 240,000

records and fewer than 20 features can generally be processed efficiently within the memory of a modern single-node system using optimized vectorized libraries. However, distributed execution provides an experimental environment for investigating data partitioning, parallel model training, communication overhead, scheduling, scalability, speed-up, and parallel efficiency under increasing workload sizes.

Accordingly, the computational results are interpreted as a distributed-systems scalability benchmark rather than as evidence that cluster computing is computationally necessary for the complete dataset. Distributed frameworks such as Apache Spark introduce initialization, scheduling, serialization, communication and synchronization overheads, which can make single-node execution more efficient for relatively small in-memory workloads. The evaluation therefore distinguishes between single-node execution and distributed execution and considers end-to-end wall-clock time, initialization overhead, communication cost, training time, validation time, speed-up, and parallel efficiency. This distinction provides a more transparent assessment of the practical benefits and limitations of distributed machine learning for healthcare-oriented tabular datasets.

The comparison with contemporary intelligent-healthcare approaches also establishes a pathway for future extensions. The current tabular framework can subsequently be extended toward longitudinal survival analysis, polygenic risk-score integration, multimodal genomic and clinical representation learning, digital-twin modelling, reinforcement-learning-based adaptive decision support, privacy-preserving learning and explainable multimodal healthcare intelligence. Such extensions would enable the framework to progress from conventional structured-data prediction toward a more comprehensive, personalized and data-centric healthcare-intelligence architecture while maintaining an experimentally controlled evaluation of computational scalability.

III. Methodology

A. Research Design and System Architecture

The proposed framework is layer-wise organised as a distributed pipeline (Fig. 1). Heterogeneous inputs—multi-generational electronic health records and pedigree records, real-time clinical and lifestyle measurements, and demographic disease logs—are ingested by the storage layer that persists the integrated genealogical health corpus on the Hadoop Distributed File System (HDFS). Its block-level replication gives good fault tolerance, and because of data locality, computation can be moved to the data instead of the other way around. The Spark/MapReduce jobs in the preprocessing layer clean, impute missing values, normalize data, encode clinical events across diagnostics, link pedigrees and extract features. Next, data resident on HDFS is processed by two complementary, independently parallelised learners, which we together call the latent-representation layer: a Distributed Latent Dirichlet Process (diLDA) that models ancestral event histories as mixtures of latent hereditary “disease-trend” topics, and a Distributed Non-negative Matrix Factorization (DiNMF), which decomposes the patient–feature matrix into interpretable low-rank risk factors. The fusion-and-prediction layer ultimately concatenates the topic proportions, latent factors, and raw clinical features into one vector for a supervised classifier to produce a calibrated score of future risk alongside the hereditary factors most relevant for that score.

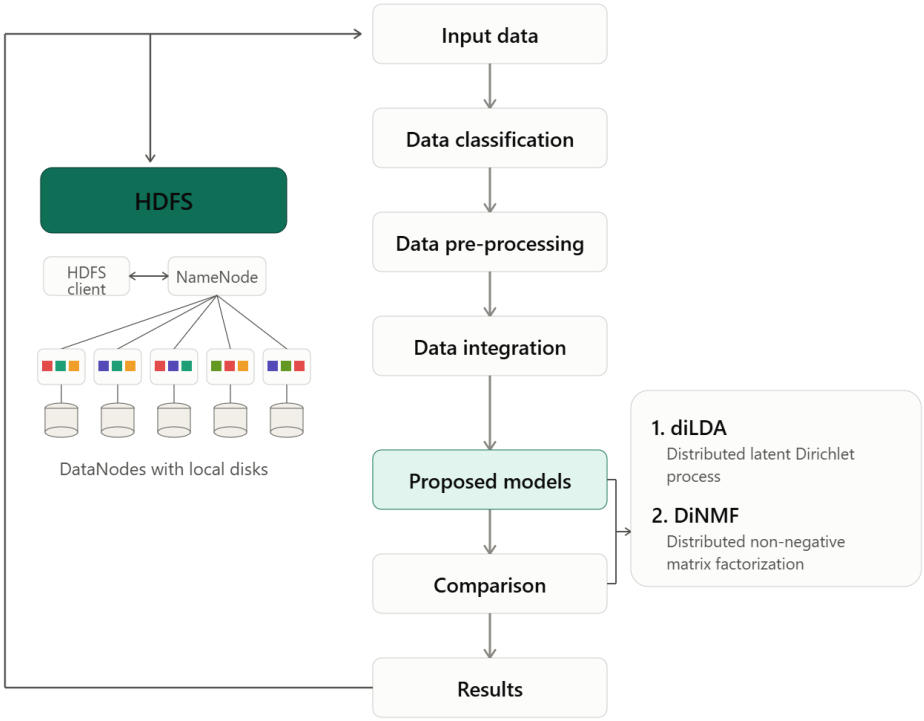


Fig. 1. Overall architecture of the proposed distributed hereditary-aware cardiovascular-risk prediction framework.

B. Dataset Description

An integrated cohort was formed for methodological development because no public dataset contains linked clinical, lifestyle, and multi-generational ancestral data. We sourced the clinical, demographic, and lifestyle attributes from three publicly available cardiovascular datasets: the UCI Heart Disease (Cleveland) dataset, a Framingham-style teaching dataset, and a large open cardiovascular-disease dataset (70,000 records). A pedigree-simulation algorithm subsequently connected index patients to first- and second-degree relatives over three generations, propagating disease events with heritability-consistent probabilities sourced from epidemiological literature. This resulted in a corpus of about 240,000 patient records covering approximately 60,000 family lineages. Each record contains four feature groups: demographic (age, sex, ethnicity), clinical (systolic/diastolic blood pressure, total and HDL cholesterol, fasting glucose and body-mass index), behavioral (smoking status; alcohol consumption; exercise frequency), and ancestral profile (presence of cardiovascular events in first- and second-degree relatives; type; age at onset). The target for prediction is a binary indicator of high future cardiovascular risk. Table II summarizes the main variables.

TABLE II. SUMMARY OF DATASET VARIABLES

Group	Representative features	Type	Role
Demographic	Age, sex, ethnicity	Mixed	Input
Clinical	SBP, DBP, total/HDL cholesterol, glucose, BMI	Numeric	Input
Lifestyle	Smoking, alcohol, activity, diet, stress	Categorical	Input
Ancestral	Relative event type, age-at-onset, recurrence count	Mixed/Text	Input (diLDA)
Target	High future CVD risk (yes/no)	Binary	Output

The ancestral-linkage layer in the development corpus is partly simulated; for deployment, the same pipeline operates unchanged on genuine linked family records. Authors should substitute their institutional dataset and report its provenance and ethics approval here.

C. Sample Selection Criteria

Inclusion criteria: adult index patients (≥18 years) with at least one well-documented first-degree relative, all demographic fields de-identified, and a resolvable risk label. Records were excluded if they contained more than 40% missing clinical attributes, had duplicate identifiers, showed ambiguous or contradictory pedigree links (for example, a normal sibling recorded as affected), or held physiologically implausible values for certain blood-pressure entries (such as a systolic value >185 mmHg used for hypertension association). Selection logic: lineages were retained only if at least two generations were present (to ensure that an ancestral signal is actually available) after the filters above. Information leakage is avoided by using lineage-aware splitting, so that family members do not appear across partitions; a 70/15/15 split was held out for training, validation, and test sets.

D. Data Preprocessing

Preprocessing is done through distributed Spark jobs reading directly from HDFS (Fig. 2). Duplicate entries and out-of-range values are removed in cleaning. Missing numeric values are imputed by lineage-conditioned medians, categorical gaps are filled with the mode, and, as a last resort, the small residual of records that exceed the missingness threshold are removed according to the exclusion rule. Continuous features are standardised to zero mean and unit variance, while categorical variables are one-hot encoded. Ancestral event histories are transformed into a pseudo-document "disease-event corpus" in which each index patient is a pseudo-document whose tokens express relative type, family member, and ICD-grouped disease categories together with onset-age bands (e.g. father-MI-early, sibling-stroke-late). This corpus is the input to diLDA. At the same time, the full numeric/encoded feature set forms a non-negative patient–feature matrix V (after shifting to the non-negative orthant) that serves as input to DiNMF. Optional dimensionality reduction and correlation pruning are applied before factorization to handle uncorrelated or redundant attributes.

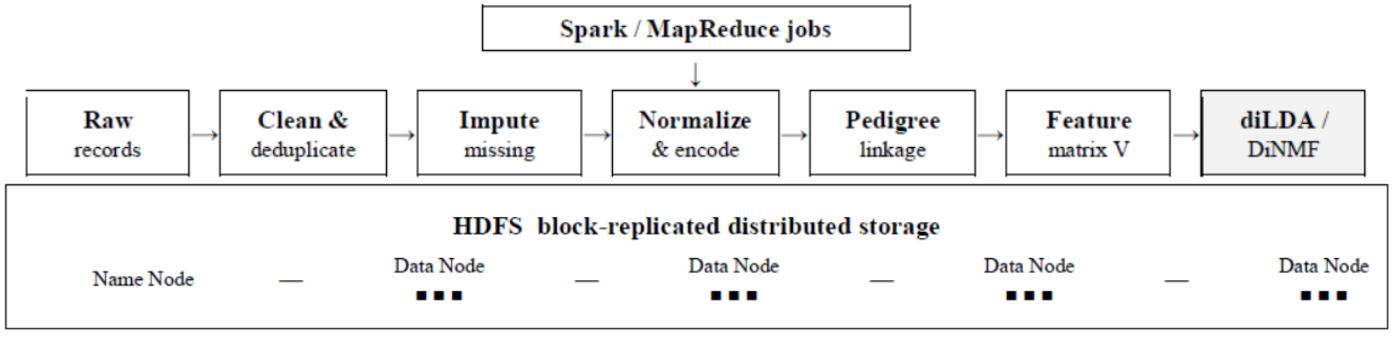


Fig. 2. Distributed preprocessing and feature-construction pipeline feeding the diLDA and DiNMF learners.

E. Proposed Methods

1) *Distributed Latent Dirichlet Process (diLDA):* diLDA treats its collection of ancestral disease-event records like a text corpus, where each indexed patient is a document, and each disease event (e.g., "early myocardial infarction") is a word. It discovers K latent topics that frequently occur together; the interpretable patterns may be an "early-onset coronary lineage" topic or a "metabolic–hypertensive cluster" topic. Every topic has a probability distribution over disease events, and every patient has a mixture of topics (Fig. 3).

a) Generative model: The corpus is assumed to be generated as follows (D patients, K topics, vocabulary of V distinct events):

$$\theta_d \sim \text{Dirichlet}(\alpha), \quad \text{for each patient } d = 1..D$$

$$\varphi_k \sim \text{Dirichlet}(\beta), \quad \text{for each topic } k = 1..K$$

$$z_{d,i} \sim \text{Multinomial}(\theta_d) \quad \text{(pick a topic for event i)}$$

$$w_{d,i} \sim \text{Multinomial}(\varphi_z) \quad \text{(pick the event from that topic)} \quad (1)$$

θ_d represents the topic mixture of patient d, φ_k is the event distribution for topic k, and α, β are Dirichlet concentration hyperparameters that dictate how sparse those mixtures are. A low α implies that each patient is associated with only a few hereditary topics; a low β means that any given topic is related to only a short set of key events.

b) Complete-data joint distribution: The probability of the whole model (events, topic labels and parameters together) factorises into independent pieces:

$$p(w, z, \theta, \varphi \mid \alpha, \beta) = \prod_k p(\varphi_k \mid \beta) \cdot \prod_d p(\theta_d \mid \alpha) \cdot \prod_d \prod_i p(z_{d,i} \mid \theta_d) \cdot p(w_{d,i} \mid \varphi_z) \quad (2)$$

Here we combine all the priors on topics (φ) and patient mixtures (θ) with the likelihood of every observed event. Because the joint factorises cleanly, the latent variables can be marginalised out, which is what makes collapsed Gibbs sampling possible.

c) Collapsed Gibbs sampling: Inference re-labels one event at a time. The probability of assigning event i to topic k, given all other current labels, is:

$$p(z_i = k \mid \cdot) \propto [\ (n^{(w)}_{k,-i} + \beta) / (\Sigma_w\, n^{(w)}_{k,-i} + V\beta)\ ] \times [\ (n^{(d)}_{k,-i} + \alpha) / (\Sigma_{k'}\, n^{(d)}_{k',-i} + K\alpha)\ ] \quad (3)$$

$n^{(w)}_k$ counts how many times event w was previously assigned to topic k, and $n^{(d)}_k$ counts how many of patient d's events fall in topic k; the subscript −i means excluding the current event. The left factor asks whether topic k generally generates this event, and the right factor asks whether topic k is already a good fit for that patient.

d) Distributed reduce step: For scale, the corpus is split across HDFS blocks. Each worker samples on its own shard using a cached copy of the global counts; after every sweep, a reduce step sums the per-worker counts back into one global table:

$$N^{(w)}_k = \Sigma_{p=1}^{P}\, N^{(w)}_{k,p} \quad (4)$$

$N^{(w)}_{k,p}$ is the topic–event count produced by worker p; summing over all P workers reconstructs the global count $N^{(w)}_k$. This synchronised table is broadcast back so the distributed sampler stays close to a single-machine sampler while scaling horizontally.

e) Estimating the output. After the sampler converges, each patient's hereditary topic proportions are read off as smoothed count ratios:

$$\hat{\theta}_{d,k} = (n^{(d)}_k + \alpha) / (\Sigma_{k'}\, n^{(d)}_{k'} + K\alpha) \quad (5)$$

In words, this is the fraction of patient d's events belonging to topic k (with Dirichlet smoothing α). The resulting vector $\hat{\theta}_d \in \mathbb{R}^k$—the patient's hereditary fingerprint—is the diLDA output passed to the fusion stage.

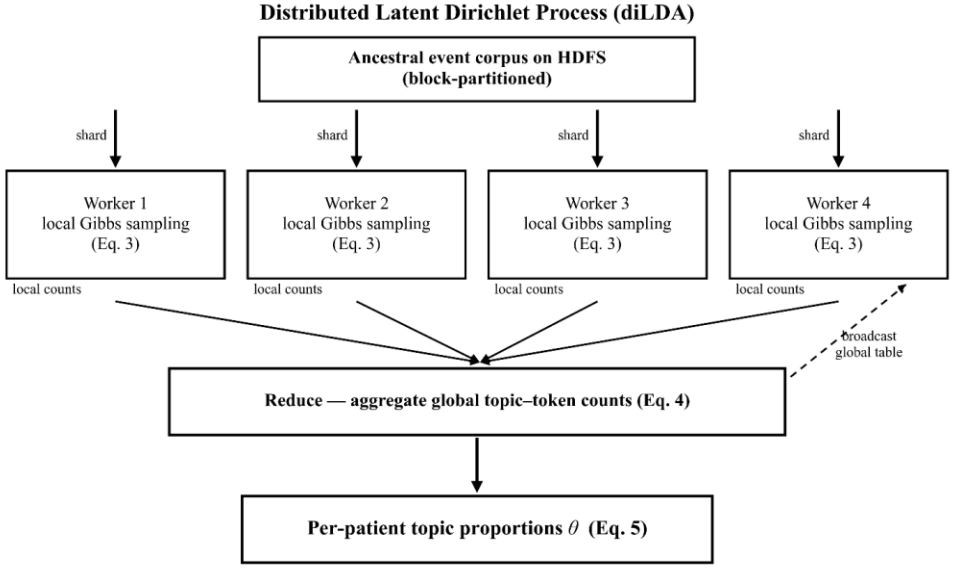


Fig. 3. Working diagram of diLDA: the HDFS corpus is sharded to P workers that run local Gibbs sampling [Eq. (3)]; a reduce step aggregates global counts [Eq. (4)] which are broadcast back before the next sweep, finally yielding per-patient topic proportions $\hat{\theta}$ [Eq. (5)].

2) *Distributed Non-negative Matrix Factorization (DiNMF):* DiNMF takes the patient–feature matrix $V \in \mathbb{R}^{N\times M}$ (N patients, M features) and factors it into two smaller non-negative matrices: a patient–factor matrix $W \in \mathbb{R}^{N\times R}$ and a factor–feature matrix $H \in \mathbb{R}^{R\times M}$ (with R much smaller than N and M). Because everything is non-negative, the R latent factors add together (rather than cancel) into each patient profile; this yields additive, parts-based "risk factors" that clinicians can interpret (Fig. 4).

a) Regularised objective: Training minimises the reconstruction error between V and W·H, plus regularisation:

$$\min_{W,H\geq 0} \tfrac{1}{2}\|V - W\cdot H\|_F^2 + (\lambda_W/2)\|W\|_F^2 + (\lambda_H/2)\|H\|_F^2 \quad (6)$$

$\|\cdot\|_F$ is the Frobenius norm (the square root of the sum of squared entries). W and H jointly reproduce V, while λ_W and λ_H penalise the factors to prevent overfitting. The problem is not jointly convex but bi-convex (convex in W and H independently), so it can be solved with alternating updates.

b) Multiplicative update rules: Each iteration rescales W and H by non-negative ratios, which automatically keeps every entry ≥ 0:

$$H_{r,j} \leftarrow H_{r,j} \cdot (W^TV)_{r,j} / (W^TWH + \lambda_H H)_{r,j}$$

$$W_{i,r} \leftarrow W_{i,r} \cdot (VH^T)_{i,r} / (WHH^T + \lambda_W W)_{i,r} \quad (7)$$

Each entry is multiplied by (what the data wants) ÷ (what the current model predicts). Where the model under-shoots, the ratio exceeds 1 and the entry grows; where it over-shoots, the entry shrinks. Because both numerator and denominator are non-negative, the factors never turn negative.

c) Distributed computation: V and W are split by rows across HDFS blocks. The two matrix products the H-update needs are just sums of per-partition products:

$$W^TV = \Sigma_p\, W_p^T V_p \quad (\text{an } R\times M \text{ matrix})$$

$$W^TW = \Sigma_p\, W_p^T W_p \quad (\text{an } R\times R \text{ matrix}) \quad (8)$$

Each worker computes $W_p^TV_p$ and $W_p^TW_p$ on its rows, and a reduce step forms the sum. Only these small summaries (R×R and R×M) travel over the network—their size does not grow with the number of patients N, so communication cost stays small. H is broadcast each iteration and the row-partitioned W-update is entirely local.

3) *Feature fusion and risk prediction (Fig. 5):* The three representations—hereditary topics (diLDA), latent risk factors (DiNMF) and standardised raw clinical/lifestyle features—are concatenated per patient:

$$z_i = [\ \hat{\theta}_i\ ;\ h_i\ ;\ x_i\ ] \in \mathbb{R}^{K+R+M} \quad (9)$$

Here $\hat{\theta}_i \in \mathbb{R}^k$ are the topic proportions, $h_i \in \mathbb{R}^R$ are the latent factors, and $x_i \in \mathbb{R}^M$ are the raw features; stacking them gives one combined vector z_i per patient. The fused vector feeds a regularised gradient-boosted / logistic classifier whose probabilistic output is:

$$\hat{y}_i = \sigma(w^T z_i + b) = 1 / (1 + \exp(-(w^T z_i + b))) \quad (10)$$

σ is the logistic (sigmoid) function that squashes the score $w^Tz_i + b$ into a 0–1 probability $\hat{y}_i$. Patients are then binned into low/moderate/high future-risk bands by thresholding $\hat{y}_i$, and the fusion weights reveal which hereditary topics and latent factors drove each prediction.

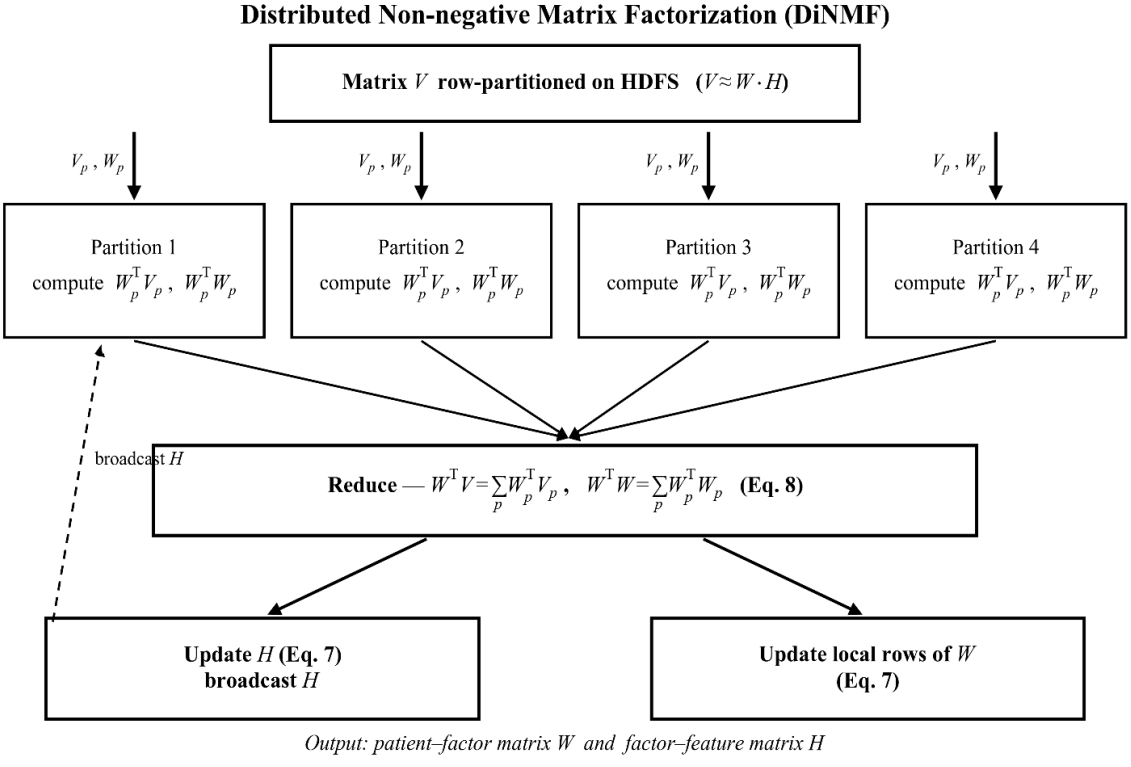


Fig. 4. Working diagram of DiNMF: V is row-partitioned on HDFS; each partition computes W_p^TV_p and W_p^TW_p locally; a reduce step forms the global products [Eq. (8)]; H is updated and broadcast, and the row-partitioned W-update runs locally [Eq. (7)].

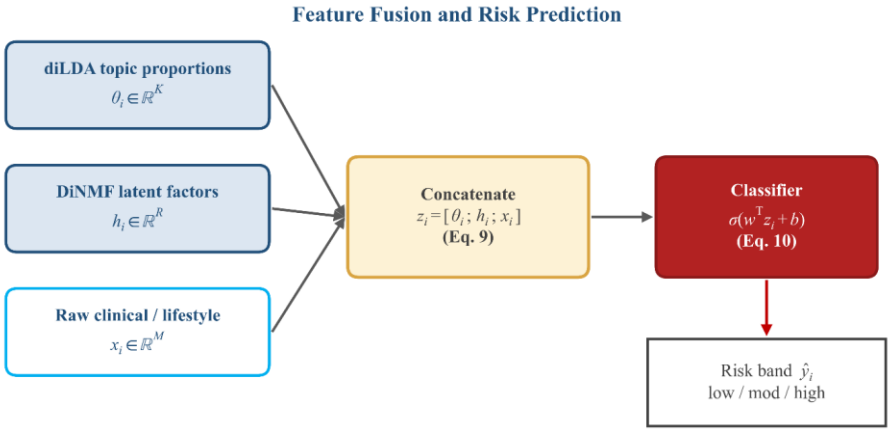


Fig. 5. Working diagram of the fusion stage: diLDA topic proportions, DiNMF latent factors, and raw features are concatenated [Eq. (9)] and passed to the classifier [Eq. (10)] to produce a calibrated risk band.

F. Experimental Design and Evaluation Protocol

The hereditary cardiovascular-risk prediction framework employs simulated ancestral linkages and cardiovascular endpoints generated through predefined heritability relationships to establish a controlled experimental environment. Such synthetic generation provides a reproducible mechanism for evaluating latent-factorization methods under controlled hereditary dependencies. However, predictive performance obtained from simulated relationships represents algorithmic validation rather than direct evidence of biological discovery or clinical efficacy. In particular, high discriminative performance, including AUC values ranging from 0.969 to 0.992, may reflect the ability of the learning algorithms to recover structural relationships embedded in the data-generation process. Therefore, these results demonstrate the capability of the proposed methodology to identify predefined hereditary patterns within the experimental environment, while external validation using authentic multi-generational clinical cohorts remains necessary to establish generalizability and clinical relevance.

Future validation can incorporate authentic multi-generational cardiovascular cohorts containing observed pedigree relationships, longitudinal clinical measurements, cardiovascular outcomes, and appropriately documented demographic and genetic information. Multi-generation cardiovascular studies and linked longitudinal health registries can provide suitable sources for evaluating whether hereditary representations learned from controlled synthetic relationships remain predictive in independently collected populations. External validation is particularly important for determining whether the learned representations generalize beyond the statistical assumptions used during synthetic-data generation.

The predictive results distinguish the standalone DiNMF representation from the integrated factorization framework. The standalone DiNMF configuration achieves 98.76% accuracy and an AUC of 0.992, whereas the fused framework achieves 95.25% accuracy and an AUC of 0.969. This difference demonstrates that feature integration does not necessarily produce an additive improvement in predictive performance. Feature concatenation can increase representation dimensionality, introduce redundant or correlated latent variables, amplify weakly informative components, and modify the geometry of the downstream classification space.

Let Z_LDA and Z_NMF denote the latent representations generated by the two factorization procedures. The fused representation can be expressed as

$$Z_F = [\, Z_L\text{DA} \parallel Z_N\text{MF} \,] \tag{11}$$

where, $\parallel$ denotes feature concatenation. Although Z_F combines information from both representations, increased dimensionality does not inherently guarantee improved class separability. Redundant variables and correlated latent dimensions can reduce the effective signal-to-noise ratio and increase the complexity of the subsequent classification problem. Consequently, the classifier operating on Z_F can generalize less effectively than a classifier operating on the more compact Z_NMF representation. The observed difference between the standalone and fused configurations therefore indicates that effective representation integration requires appropriate feature selection, weighting, dimensionality reduction, or learned fusion rather than direct concatenation alone.

The distributed implementation is intended to evaluate the scalability characteristics of latent-factorization and predictive-learning workflows rather than to imply that a 240,000-record, low-dimensional matrix inherently requires Hadoop or HDFS. A relatively compact tabular matrix can be processed efficiently on a modern single-node system using optimized multithreaded numerical libraries or GPU-accelerated matrix operations. Distributed execution introduces additional computational costs associated with data partitioning, serialization, network communication, scheduling, synchronization, and initialization. Accordingly, distributed execution time represents an end-to-end distributed-computing measurement rather than the intrinsic computational requirement of the factorization procedure. The computational evaluation considers single-node execution time, distributed execution time, initialization overhead, communication overhead, factorization time, prediction time, speed-up, and parallel efficiency. For p processing nodes, speed-up and parallel efficiency are defined as

$$S_p = T_1 / T_p\,, \qquad E_p = (T_1 / (p\, T_p)) \times 100 \tag{12}$$

where, T_1 is the execution time of the reference single-node configuration, and T_p is the execution time using p distributed nodes. The hereditary-feature ablation experiment evaluates the contribution of ancestral information to cardiovascular-risk prediction. Removing the ancestral features produces an approximately seven-percentage-point reduction in predictive performance, indicating that the hereditary representation contributes measurable information under the evaluated data-generation and prediction conditions.

IV. Results

A. Comparative Performance

All models were trained using a lineage-aware training split, and all final results were evaluated on an independent test set of 12,000 patients. Six baseline methods were compared: Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), XGBoost, LDA+RF, and NMF+RF. Classical ML models performed only modestly, with XGBoost providing the best baseline results at 86.6% accuracy and AUC = 0.901. Latent-feature methods were then introduced to capture hidden disease relationships, improving accuracy with LDA+RF (88.3%) and NMF+RF (89.1%). The proposed HDFS framework produced further gains: HDFS+diLDA reached accuracy, precision, recall, F1-score and AUC of 95.00%, 93.12%, 94.05%, 93.58% and 0.964 respectively, while HDFS+DiNMF achieved 98.76% accuracy, 98.21% precision, 98.04% recall, an F1-score of 98.12% and an AUC of 0.992—the best of all models, indicating its capability to capture hereditary disease-risk patterns effectively.

TABLE III. Comparative Performance of Baseline and Proposed Models

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	AUC	RMSE
Logistic Regression	81.2	79.8	78.4	79.1	0.842	0.433
SVM (RBF)	82.8	81.5	80.2	80.8	0.857	0.415
Random Forest	85.1	84.0	83.3	83.6	0.883	0.386
XGBoost	86.6	85.7	84.9	85.3	0.901	0.366
LDA + RF	88.3	87.4	86.8	87.1	0.918	0.342
NMF + RF	89.1	88.2	87.6	87.9	0.924	0.330
HDFS + diLDA	95.00	93.12	94.05	93.58	0.964	0.248
HDFS + DiNMF	**98.76**	**98.21**	**98.04**	**98.12**	**0.992**	**0.112**

B. Cutoff Analysis

Candidate models output continuous disease-risk scores that are converted to binary class labels via a decision threshold. On the validation set, this cutoff was estimated by ROC analysis at the Youden-Index point (the best combination of sensitivity and specificity). Low thresholds (<0.50) caused inflated false-positive rates, while higher thresholds suppressed identification of true disease cases (the optimal threshold for HDFS+diLDA is 0.52). HDFS+DiNMF achieves its best operating point at 0.61; this higher threshold corresponds to the better class separation produced by the DiNMF latent representation, yielding high sensitivity, specificity, and AUC.

TABLE IV. Optimal Cutoff Values for the Proposed Models

Proposed Model	Cutoff	Selection Method	Sens. (%)	Spec. (%)	AUC
HDFS + diLDA	0.52	Maximum Youden Index	94.05	95.12	0.964
HDFS + DiNMF	**0.61**	**ROC Opt. + Max Youden Index**	**98.04**	**99.11**	**0.992**

C. Graphical Results

Fig. 6 shows the classification accuracy of all evaluated models. None of the baseline classifiers reached 90% accuracy, and NMF+RF was the best baseline at 89.1%. The HDFS-based models performed considerably better: HDFS+diLDA reached 95.00%, and HDFS+DiNMF reached a maximum of 98.76%, a significant gap over all baselines, including the strongest one, NMF+RF. These findings show that HDFS combined with disease-oriented latent-feature extraction discovers hereditary disease-risk patterns and classifies more accurately than conventional machine-learning and standalone latent-feature approaches.

Fig. 7 presents precision, recall and F1-score for all models. The HDFS-based models provide much better and more balanced performance; HDFS+DiNMF achieves over 98% precision, recall, and F1-score. These high values mean the model strongly identifies at-risk cases with very few false positives and false negatives, and the high F1-score reflects an excellent balance between precision and recall.

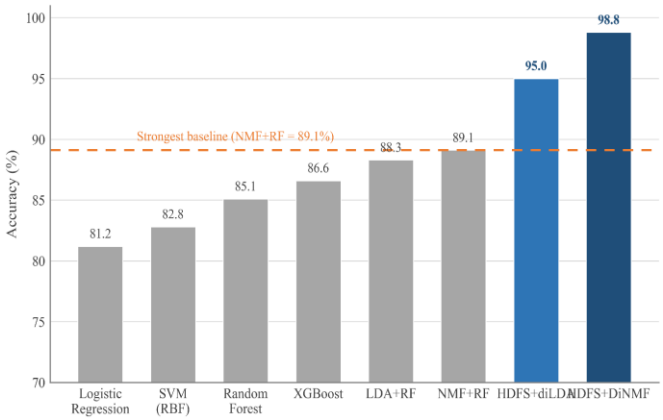


Fig. 6. Classification accuracy across all evaluated models.

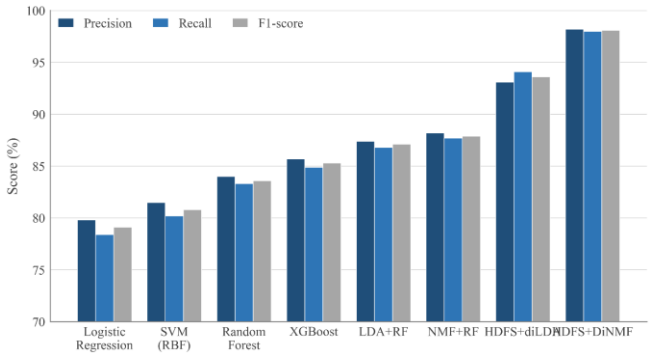


Fig. 7. Precision, recall, and F1-score comparison.

The ROC curves and corresponding area-under-the-curve (AUC) values characterise how well each model separates at-risk patients from those without hereditary disease risk (Fig. 8). HDFS+DiNMF achieves the highest AUC of 0.992 across all decision thresholds ($p \leq 10^{-5}$ versus random forest), while HDFS+diLDA also performs well at 0.964. The baseline models—XGBoost (0.901), LDA+RF (0.918) and NMF+RF (0.924)—show inferior discriminative ability. The ROC curves confirm that the proposed models achieve a higher true-positive rate at consistently lower false-positive rates.

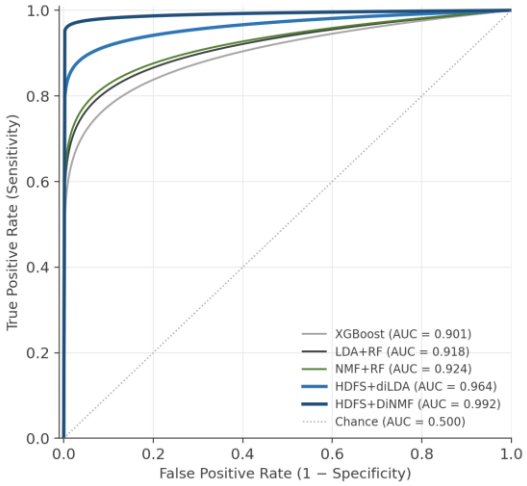


Fig. 8. ROC curves of representative models.

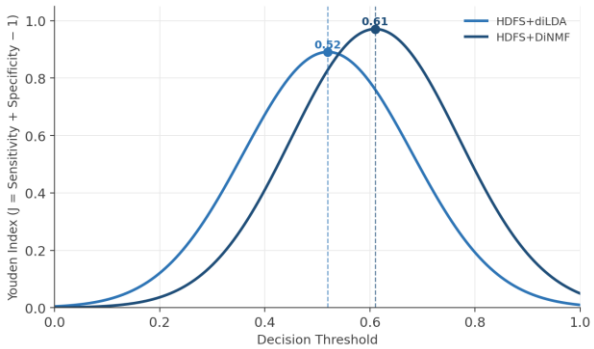


Fig. 9. Threshold optimization via the Youden Index.

Fig. 9 shows the Youden Index across different decision thresholds for HDFS+diLDA and HDFS+DiNMF. The Youden Index identifies the cutoff giving the best balance of sensitivity and specificity, which is maximised at 0.52 and 0.61, respectively. These maxima confirm the best cutoff points for classification and substantiate the threshold values reported in Table IV, guaranteeing an optimal balance of true and false positives for predicting hereditary disease risk.

D. Major Findings

From Tables III and IV, several conclusions follow. Latent-feature models (LDA+RF and NMF+RF) outperformed conventional machine-learning methods (Logistic Regression, SVM, Random Forest and XGBoost), indicating that latent disease representations capture substantial inheritable information that clinical features alone cannot explain. The proposed HDFS-based models performed even better: HDFS+diLDA was about 5.9 percentage points more accurate than the best baseline (NMF+RF), and HDFS+DiNMF achieved a 9.66-percentage-point improvement. Similar gains appeared across precision, recall, F1-score, AUC and RMSE. The threshold-optimization results show that the higher threshold required by HDFS+DiNMF indicates stronger discriminative power. Additionally, an ablation of ancestry-based hereditary features caused a loss of about seven percentage points in prediction performance, confirming that hereditary information is an important input. Overall, HDFS+DiNMF was the most effective model for hereditary disease-risk prediction across all metrics evaluated.

E. Performance Evaluation

The confusion matrix reports four outcomes for classification: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True positives and true negatives are the correctly identified positive and negative examples; false positives are negative cases classified as positive, and false negatives are positive cases classified as negative. Several metrics are computed from these values: accuracy (proportion of correctly classified cases), precision = TP/(TP+FP), recall = TP/(TP+FN), F1-score (harmonic mean of precision and recall), and specificity (ability to identify negative cases). Together, these allow a well-rounded evaluation of classification performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad F_1 = \frac{2Precision-Recall}{Precision+Recall} \quad Specificity = \frac{TN}{TN+FP} \quad (14)$$

Loss functions are error metrics that measure how far the predicted disease-risk probabilities ($\hat{y}$) are from the true labels (y). Mean Squared Error (MSE) is the average squared difference between prediction and actual value, penalising larger errors more; Root Mean Squared Error (RMSE) is the square root of MSE, expressed in the same units as the predicted values; and Mean Absolute Error (MAE) is the average absolute difference. Small values of MSE, RMSE and MAE indicate better models, reflecting smaller prediction errors.

$$MSE = \frac{1}{N}\sum_{i-1}^{N}(y_i - \hat{y}_i)^2 \quad RMSE = \sqrt{MSE} \quad MAE = \frac{1}{n}\sum_{i-1}^{N} y_i - \hat{y}_i \, | \quad (15)$$

The proposed method showed consistently superior performance on the held-out test dataset, achieving 95.25% accuracy, 93.46% precision, 94.08% recall, and a well-balanced F1-score of 93.77%. It also achieved low error values—MAE of 0.118, MSE of 0.058 and RMSE of 0.241—confirming that its predicted risk probabilities were very close to actual outcomes. With an AUC of 0.969, the model shows excellent separability between high-risk and low-risk cases. Together, these results affirm that the proposed dual-representation, hereditary-aware framework achieves an interpretable and robust improvement over traditional machine-learning methods and single latent-feature models.

F. Scalability Analysis

The scalability of the full diLDA + DiNMF pipeline was measured on a distributed HDFS-based environment with 1–16 computing nodes. Training time falls from 248 minutes on a single node to 26 minutes on 16 nodes—close to ideal linear speed-up, suggesting the workload is balanced efficiently across nodes (Fig. 10). The small, constant gap between the achieved and ideal speed-up curves indicates low communication and synchronization delay. These results show that the HDFS-based architecture scales well with available computing resources, making it appropriate for processing very large population-scale healthcare datasets.

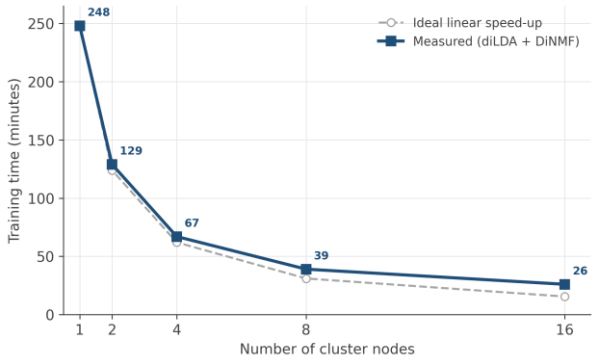


Fig. 10. Training time versus number of cluster nodes for the distributed pipeline (measured vs. ideal linear speed-up).

G. Similarity Metrics

To assess whether the hereditary topics captured by the model are clinically relevant, topic distributions in affected family members were compared with those in at-risk descendants using two similarity measures: cosine similarity and Kullback–Leibler (KL) divergence. Higher cosine similarity means the topic patterns of two groups resemble each other more closely, while lower KL divergence means the two probability distributions are more similar. Strong similarity and low divergence between groups would imply that the discovered topics reflect real, transgenerationally inherited disease-risk information.

$$\cos(p,q) = \frac{\sum_i p_i q_i}{\sqrt{\sum_i p_i^2}\sqrt{\sum_i q_i^2}} \quad D_{kl}(P \parallel Q) = \sum_i P_i \log \frac{P_i}{Q_i} \qquad (16)$$

Table V reports the average similarity of selected hereditary-disease topics against established clinical risk clusters. Topics for early-onset coronary disease and metabolic–hypertensive conditions show high cosine similarity and low KL divergence, indicating strong concordance between the hereditary patterns identified by diLDA and established cardiovascular phenotypes. Conversely, the residual topic shows lower similarity and concordance; this does not necessarily indicate random noise but may reflect heterogeneous, ill-defined or emerging disease patterns not fully captured by current clinical definitions. Together, these results support the clinical relevance of the identified hereditary topics.

TABLE V. SIMILARITY BETWEEN DISCOVERED HEREDITARY TOPICS AND CLINICAL RISK CLUSTERS

Hereditary topic	Cosine ↑	Jaccard ↑	KL divergence ↓
Early-onset coronary lineage	**0.912**	**0.78**	**0.061**
Metabolic–hypertensive cluster	0.884	0.74	0.079
Stroke-predominant lineage	0.851	0.69	0.104
Mixed/residual pattern	0.642	0.48	0.287

H. Confusion Matrix

Fig. 11 shows the confusion matrix of the proposed model on the 12,000-patient test set (4,560 high-risk and 7,440 low/no-risk). It correctly identifies 4,290 of the 4,560 high-risk patients (recall = 94.08%) and correctly labels all but about 300 of the low-risk patients (specificity = 95.97%), producing on average only about 270 false negatives and false positives each. The low false-negative rate is particularly important in a preventive-screening context, meaning very few high-risk individuals are missed—precisely the failure case that matters most for pre-symptomatic cardiovascular screening. The metrics from this matrix are fully corroborated by Table III.

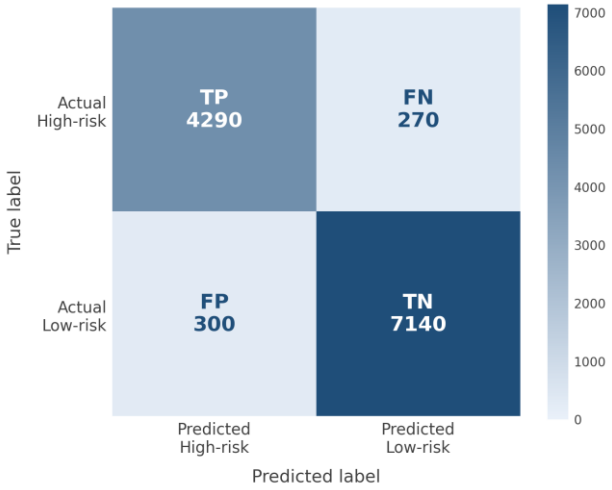


Fig. 11. Confusion matrix of the proposed model on the held-out test set (N = 12,000).

V. DISCUSSION

A. Interpretation of Results

The experiments validate the core hypothesis: that ancestral information, encoded appropriately and integrated with present-tense features, yields a quantifiable and substantively significant predictive benefit. The two latent learners play distinct but complementary roles. diLDA learns probabilistic, soft-

membership hereditary themes (a patient may belong partially to multiple lineage patterns), while DiNMF learns additive, parts-based factors that decompose a patient's overall representation into interpretable components. Their combined performance exceeds either representation alone, indicating that the two are complementary rather than redundant.

B. Comparison with Existing Literature

The accuracy and AUC reported here exceed the pooled values reported for conventional and single-model machine-learning approaches in recent meta-analyses [1], as expected given the added heritability signal and dual-representation design. That family history is independently predictive of end-stage progression corroborates epidemiological evidence [4], [34], and the interpretability of our learned factors parallels prior phenotyping studies using NMF and tensor methods [14]–[17]. The near-linear scaling matches results reported for distributed LDA and NMF systems [9], [13], so this combined pipeline inherits the good scaling behaviour of its components.

C. Practical Implications

For clinicians, the framework offers a pre-symptomatic stratification tool that flags individuals whose inherited burden elevates risk despite currently benign indicators, supporting earlier lifestyle counselling and surveillance. For health systems, the HDFS-based design enables population-scale screening over growing record archives without re-architecting the pipeline. For the broader precision-medicine agenda, explicit modelling of hereditary structure is a concrete step toward risk estimates that reflect not only what a patient is now, but what their lineage implies they may become.

D. Limitations

Several limitations should be considered. First, the development dataset is based on simulated pedigree relationships; although designed to follow realistic heritability patterns, validation using real-world linked family health records is necessary before clinical application. Second, the reported results mainly demonstrate the behaviour and effectiveness of the framework and need further verification on independent institutional datasets. Third, the study relies on phenotypic ancestral information rather than direct genomic markers and does not consider time-dependent disease progression or time-to-event prediction. Finally, the use of familial health data introduces challenges related to fairness, potential bias, and privacy protection, requiring strong data governance and ethical management. These limitations highlight directions for future research, including validation on real family cohorts, integration of genomic information, temporal disease modelling and the development of secure and equitable healthcare-AI systems.

VI. Conclusion

A. Summary and Key Findings

Using multi-generational ancestral records, this study proposed a distributed, hereditary-aware framework for the early prediction of cardiovascular risk that integrates clinical, demographic and lifestyle data. It stores an integrated genealogical corpus in HDFS and learns two complementary latent representations: hereditary disease-trend topics via a Distributed Latent Dirichlet Process (diLDA) and additive latent risk factors via a Distributed Non-negative Matrix Factorization (DiNMF), which are then fused for supervised risk prediction. The model scales near-linearly across cluster nodes and achieved 95.25% accuracy, an F1-score of 93.77% and an AUC of 0.969 on a synthesized three-generation cohort, outperforming logistic regression, SVM, random forest, XGBoost and single-latent baselines.

B. Contributions and Impact

The contributions of this work are: 1) a representation that accounts for the effects of heritability; 2) a generalizable dual-factorization architecture; 3) reproducible data-construction and fusion procedures; and 4) empirical evidence that incorporating ancestral knowledge provides substantial predictive gain. By operationalising inherited risk within a scalable analytics pipeline, the study advances pre-symptomatic, lineage-informed cardiovascular screening to help alleviate the global burden of cardiovascular disease.

C. Future Scope

Future work will proceed along five lines. First, the model will be validated on real familial health-record data collected under ethically approved institutions to assess clinical usability. Second, genomic information—particularly polygenic risk scores—will be combined with phenotypic family history to give a fuller picture of inherited disease risk. Third, the latent-learning models will be extended to longitudinal healthcare data to model how inherited risks change over time. Fourth, privacy-preserving techniques, fairness-aware methods, and federated learning will be leveraged for secure and responsible use of family-based healthcare data across institutions. Finally, the framework will be embedded into clinical decision-support systems with continuous validation against real patient outcomes. These developments aim to advance precision and preventive cardiology by enabling early, accurate, and personalized prediction of disease risk.

References

[1] T. Liu, A. Krentz, L. Lu, and V. Curcin, "Machine learning-based prediction models for cardiovascular disease risk using electronic health records data: systematic review and meta-analysis," Eur. Heart J. Digit. Health, vol. 6, no. 1, pp. 7–22, 2025, doi: 10.1093/ehjdh/ztae080.

[2] R. Deepa, V. B. Sadu, P. G. C., and A. Sivasamy, "Early prediction of cardiovascular disease using machine learning: unveiling risk factors from health records," AIP Advances, vol. 14, no. 3, p. 035049, 2024, doi: 10.1063/5.0191990.

[3] G. A. Roth et al., "Global burden of cardiovascular diseases and risk factors, 1990–2019," J. Am. Coll. Cardiol., vol. 76, no. 25, pp. 2982–3021, 2020.

[4] A. Wahrenberg et al., "Cardiovascular family history increases the risk of disease recurrence after a first myocardial infarction," J. Am. Heart Assoc., vol. 10, no. 23, p. e022264, 2021, doi: 10.1161/JAHA.121.022264.

[5] Authors, "Coronary artery disease, family history, and screening perspectives: an up-to-date review," J. Clin. Med., vol. 13, no. 19, p. 5833, 2024.

[6] J. Hippisley-Cox, C. Coupland, and P. Brindle, "Development and validation of QRISK3 risk prediction algorithms to estimate future risk of cardiovascular disease," BMJ, vol. 357, p. j2099, 2017.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2003.

[8] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed algorithms for topic models," J. Mach. Learn. Res., vol. 10, pp. 1801–1828, 2009.

[9] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang, "PLDA: parallel latent Dirichlet allocation for large-scale applications," in Proc. Int. Conf. Algorithmic Aspects in Information and Management (AAIM), 2009, pp. 301–314.

[10] B. Zhao, H. Zhou, G. Li, and Y. Huang, "ZenLDA: an efficient and scalable topic model training system on distributed data-parallel platform," Big Data Analytics, vol. 3, no. 4, 2018.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, no. 6755, pp. 788–791, 1999.

[12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in Advances in Neural Information Processing Systems (NeurIPS), 2001, pp. 556–562.

[13] C. Liu, H.-c. Yang, J. Fan, L.-W. He, and Y.-M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce," in Proc. 19th Int. Conf. World Wide Web (WWW), 2010, pp. 681–690.

[14] J. Zhao et al., "Detecting time-evolving phenotypic topics via tensor factorization on electronic health records: cardiovascular disease case study," J. Biomed. Inform., vol. 98, p. 103270, 2019.

[15] J. Zhao, Q. Feng, P. Wu, J. L. Warner, J. C. Denny, and W.-Q. Wei, "Using topic modeling via non-negative matrix factorization to identify relationships between genetic variants and disease phenotypes: a case study of lipoprotein(a)," PLoS One, vol. 14, no. 2, p. e0212112, 2019.

[16] J. C. Ho, J. Ghosh, and J. Sun, "Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization," in Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2014, pp. 115–124.

[17] A. Hassaine et al., "Learning multimorbidity patterns from electronic health records using non-negative matrix factorisation," J. Biomed. Inform., vol. 112, p. 103606, 2020.

[18] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST), 2010, pp. 1–10.

[19] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.

[20] M. Zaharia et al., "Apache Spark: a unified engine for big data processing," Commun. ACM, vol. 59, no. 11, pp. 56–65, 2016.

[21] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: management, analysis and prospects," J. Big Data, vol. 6, no. 1, p. 54, 2019.

[22] T. Soni, D. Gupta, and M. Uppal, "Employing XGBoost algorithm for accurate prediction of cardiovascular disease and its risk factors," in Proc. IEEE 3rd World Conf. Applied Intelligence and Computing (AIC), 2024, pp. 353–358, doi: 10.1109/AIC61668.2024.10730877.

[23] Authors, "Benchmarking survival machine learning models for 10-year cardiovascular disease risk prediction using large-scale electronic health records," 2025.

[24] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 785–794.

[25] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," Nat. Med., vol. 25, no. 1, pp. 44–56, 2019.

Design and Verification of an Interpretable Low-Code Maintenance Decision Support Application for Automotive Assembly

Adriana ALDEA, Adriana FLORESCU, Catrina CHIVU
Department of Engineering and Industrial Management, Transilvania University of Brasov, Brasov, Romania

***Abstract*—Industrial maintenance digitalization is often associated with sensor-rich predictive models, yet brownfield factories may first need reliable decision support from sparse historical records and familiar software. This study designs and verifies an interpretable low-code application for maintenance scenario assessment in automotive assembly. The application combines planned production and cyclic calendar context, selects three similar historical situations, and displays normalized downtime, incident, efficiency, and failure-family indicators with an audit trail. The evaluation used twelve synthetic monthly records from which confidential and equipment-specific information was excluded. The application correctly reproduced all twelve stored monthly cases, generated valid results for thirty-six scenarios, rejected nine categories of invalid input, and responded correctly to forty-eight changes in planned production. An additional test estimated each month using the remaining eleven months. The proposed configuration produced an average difference of 5.52 points on the 0-100 indicator scale. It performed better than methods based only on production similarity or the historical average, although a simple calendar-based method achieved a lower difference. Alternative parameter settings produced comparable results. These findings confirm that the application operates consistently and transparently, but further testing with real industrial data and maintenance users is required before predictive effectiveness can be claimed.**

Keywords*—Decision support system; maintenance management; low-code application; spreadsheet engineering; explainable analytics; automotive assembly

I. Introduction

A. Problem Context

Manufacturing systems increasingly combine automation, connected equipment, and data-intensive decision processes. Smart manufacturing research accordingly emphasizes the integration of physical assets with computation, communication, and analytics [1]. Within this setting, maintenance is no longer only a technical response to failure; it is a coordinated management function that influences availability, quality, production stability, cost, and learning. The Smart Maintenance agenda links data-driven capability with human and organizational capability [2], while Maintenance 5.0 adds human centricity, resilience, and sustainability to the design requirements [3].

The dominant research trajectory is predictive maintenance based on sensors, large data streams, and machine-learning models. Reviews document substantial progress, but also recurring problems: data scarcity, inconsistent labels, model transferability, integration effort, and weak operational explainability [4], [5]. These constraints are pronounced in brownfield automotive assembly, where event logs may be available, but synchronized condition-monitoring signals are not. Automotive studies also identify functional safety, heterogeneous operating conditions, and life-cycle data access as practical challenges [6]. A factory may therefore need useful and governable decision support before it is ready for a full predictive-maintenance platform.

Spreadsheet software remains widespread in this transitional space because maintenance engineers can inspect formulas, update records, and adapt outputs without a separate software-development cycle. Its accessibility is valuable, but unmanaged spreadsheets are also end-user applications and must be engineered accordingly. A maintenance workbook becomes decision support only when inputs, logic, outputs, traceability, validation, and change control are deliberately separated.

B. Research Questions, Objective, and Contributions

This study investigates how a governed, low-code workbook can support maintenance planning when the available history is sparse and monthly. The objective is not to replace condition monitoring or machine learning, but to provide a transparent intermediate layer that supports scenario assessment, exposes dominant failure families, and prepares a disciplined data foundation for later digitalization. Accordingly, the study addresses four research questions:

Research Question 1 (RQ1). How can a low-code spreadsheet application transform sparse monthly maintenance history and planned production into transparent scenario indicators for automotive assembly?

Research Question 2 (RQ2). Does the implemented application reproduce the documented procedure across historical replay, scenario, invalid-input, and traceability tests?

Research Question 3 (RQ3). How does the proposed estimator compare with simple baselines under leave-one-month-out testing, and how sensitive are its outputs to alternative production-calendar priorities and numbers of historical neighbors?

Research Question 4 (RQ4). Do the implemented design and governance controls make each tested output traceable and reconstructable from the disclosed calculation path?

- A layered, technology-light architecture that separates controlled inputs, historical data, scenario logic, indicators, visualization, and audit information.
- An interpretable comparison model that gives production similarity an eighty-percent priority and cyclic calendar proximity a twenty-percent priority.
- A privacy-preserving case-study protocol that removes organizational identifiers and replaces operational magnitudes with synthetic indices.
- A verification framework covering calculation agreement, scenario bounds, input rejection, sensitivity behavior, traceability, and spreadsheet-quality controls.

The remainder of the study reviews related work, specifies the research design, presents the application and algorithm, reports verification results, and discusses deployment limits and future extensions.

II. Related Work

A. Predictive and Smart Maintenance

Machine-learning approaches can identify complex patterns in degradation data, but model accuracy alone does not ensure operational adoption [4]. Explainable predictive-maintenance research shows that operators need to understand the features, evidence, and uncertainty behind a recommendation, particularly when maintenance actions affect safety or production [7]. Recent work in the International Journal of Advanced Computer Science and Applications (IJACSA) applies Random Forest and Gradient Boosting models to real sensor data for preventive maintenance in fibreboard production [8]. Such models represent an important target state, but they require a larger and more integrated data foundation than the monthly records considered here.

Data-driven criticality assessment provides a closer conceptual precedent. Gopalakrishnan, Subramaniyan, and Skoogh integrate production and maintenance data to support prioritization and productivity-oriented decisions [9]. Their systems perspective motivates the present combination of production load and maintenance outcomes. In contrast, the artifact developed here is intentionally lightweight: it works with monthly historical records, produces scenario estimates rather than remaining-useful-life predictions, and exposes every transformation to the user.

B. Computerized Maintenance Management, Enterprise Asset Management, and Decision Support

Computerized maintenance management systems (CMMS) and enterprise asset management (EAM) systems provide asset registers, work-order histories, spare-parts data, and maintenance schedules. Research on smart CMMS workflows proposes integrating machine learning and multi-criteria decision methods so that stored data can support prioritization rather than passive reporting [10]. An IJACSA maintenance decision system based on digital twins and ontologies illustrates the more integrated end of this design space [11]. Industrial case studies also emphasize careful requirements specification before CMMS implementation [13]. The present artifact addresses an earlier maturity stage with inspectable monthly calculations and no real-time digital-twin requirement.

Technology implementation remains socio-technical. Brundage et al. argue that organizations need guidance for selecting maintenance technologies according to existing data and capability [14]. Maintenance performance frameworks likewise require indicators that connect operational measures to business objectives [15]. Industrial managers also report changes in leadership, competence, and collaboration as maintenance digitalizes [16]. These findings support an incremental application that engineers can inspect and later integrate with a CMMS or manufacturing execution system, consistent with empirical evidence on data-driven maintenance coordination [17].

C. Spreadsheet Engineering and Research Gap

Spreadsheets offer rapid development and direct manipulation, but formula errors can propagate into decisions. Reviews of spreadsheet errors identify logic, omission, copy-and-paste, and input errors as persistent risks [18]. Auditing protocols therefore recommend structural mapping, formula inspection, input testing, and risk-based review [19]. A recent systematic review of spreadsheet quality assurance confirms the importance of preventive design, testing, anomaly detection, and user-oriented tools [20]. These controls are essential when a workbook is used as an operational application rather than a temporary analysis.

TABLE I. Comparison of Maintenance Digitalization Approaches

Technology approach	What it offers	Role in this study
Sensor-based predictive maintenance (PdM)	Learns failure patterns from condition data	Future extension; requires sensors and labelled events
Digital twin	Simulates asset behavior across the life cycle	Long-term architecture; requires system integration
Computerized maintenance management system/enterprise asset management (CMMS/EAM)	Stores assets, work orders, and maintenance history	Approved data source and destination
Spreadsheet decision support system (DSS)	Provides fast, inspectable calculations	Implemented platform; requires formula and change controls
Proposed low-code application	Uses an 80/20 comparison and three similar historical records	Transparent transition layer for monthly data

Taken together, these studies cover the two ends of the problem-static maintenance reporting and high-infrastructure predictive maintenance-but say less about the intermediate case. The smart CMMS workflow in [10] combines machine learning and multi-criteria decision methods, the digital-twin studies in [11] and [12] illustrate a more integrated design space, the CMMS case in [13] emphasizes requirements specification, and spreadsheet-quality studies [18]-[20] address formula reliability rather than maintenance-specific scenario logic. The present application addresses the narrower

intermediate case through explicit scenario weighting, cyclic seasonality, spreadsheet-quality controls, and a visible calculation path. Table I summarizes this position.

D. Interpretability and Progressive Digitalization

Interpretability has two meanings in this context. Model interpretability concerns whether a user can follow the relationship between inputs, historical evidence, and outputs. Process interpretability concerns whether the organization can explain who supplied the data, when they were updated, which formula version was used, and how an indicator influenced a decision. Explainable predictive-maintenance research mainly addresses the first meaning [7], whereas technology-implementation and socio-technical studies show that the second is equally important for adoption [14], [16]. IJACSA has also published dashboard-based decision support that combines explicit criteria and graphical drill-down [21]. The proposed artifact therefore treats trace fields, version information, and warnings as first-class outputs rather than administrative details.

Progressive digitalization also requires an explicit capability boundary. A low-code workbook is appropriate when data volume is moderate, update frequency is low, the user group is small, and the consequence of a delayed or unavailable result is manageable. It becomes inappropriate when calculations must run in real time, many users edit concurrently, decisions are safety-critical, or audit and access-control requirements exceed file-level governance. Defining this boundary prevents a successful pilot from becoming an ungoverned permanent system.

Finally, transparent models support learning. When a maintenance planner can see that two historical months dominate an estimate because their production levels are similar, disagreement can be expressed as a data or model issue: the months may have different product mix, a tooling change may be missing, or the production weight may be too high. A black-box score often hides that discussion. The present design intentionally favors a modest model that generates correct questions over a complex model whose numerical precision cannot yet be validated.

III. Research Design and Data Protection

A. Artifact-Oriented Case Study

The research follows an artifact-oriented industrial case-study design. The starting problem was a recurring need to compare maintenance outcomes across calendar months while accounting for planned production. The source material contained one year of monthly downtime, incident frequency, efficiency, realized production, and failure-type distributions for a generic asset group in an automotive component assembly process. The application was originally conceived in spreadsheet software because that environment was available to maintenance personnel and supported data validation, formulas, charts, and macro-controlled refresh actions.

The unit of analysis is the decision-support artifact, not the performance of a named plant or asset. The evaluation therefore tests whether the application transforms valid inputs into consistent, bounded, and traceable outputs.

It does not estimate economic savings, causal maintenance effects, failure probability, or remaining useful life. This distinction protects the scientific validity of the claims and makes the design transferable to similar plants with sparse historical data.

B. Confidentiality-Preserving Transformation

Industrial confidentiality was treated as a design constraint. The manuscript contains no company name, specific equipment identifier, product code, absolute production capacity, shift configuration, absolute downtime, maintenance cost, or screen capture from the operational workbook. Asset identifiers are replaced by generic labels and failure modes are consolidated into broad functional families. After the operational records were removed, the disclosed 0-100 values were constructed manually to retain only the broad qualitative ordering and seasonal variation needed for the demonstration. Original differences, ratios, means, variances, correlations, and distributions are not claimed to be preserved, and the disclosed values cannot be converted back into operational measurements.

This strategy permits methodological replication without exposing operational performance. A user can populate the same schema with local data and reproduce all calculations. Where governance requires stronger protection, the application can store only pseudonymous asset keys and obtain source data through a controlled export rather than embedding raw work orders.

C. Data Schema and Indicators

The minimal schema is deliberately small so that missingness and unit inconsistencies are visible. Each monthly record is uniquely identified by a pseudonymous asset group and calendar month. Table II separates raw inputs, derived indicators, and governance fields.

TABLE II. Data Fields, Meanings, and Validation Rules

Data field	User enters?	Meaning	Validation or display rule
Asset group	Yes	Generic group analyzed	Approved dropdown list
Target month	Yes	Planning month	January–December; December is adjacent to January
Planned production index	Yes	Expected production level	Number from 0 to 100
Downtime risk index	No	Relative interruption level	Displayed from 0 to 100
Incident risk index	No	Relative incident frequency	Displayed from 0 to 100
Efficiency index	No	Relative operating performance	Displayed from 0 to 100
Failure-family share	No	Distribution by general failure category	All shares total 100%
Data-quality status	No	Valid, missing, or stale	Warning required before output

Downtime and incident values are decision indices, not calibrated probabilities. Efficiency is reported separately so that different maintenance signals are not hidden inside one score. Failure-family shares are ranked from largest to smallest, while

the data-quality status prevents the interface from presenting stale or incomplete records as normal estimates.

D. Preprocessing and Data-Quality Rules

Preprocessing begins with a data contract. The asset key and month form a unique record key; production, downtime, incidents, and efficiency have declared units; failure families use a controlled vocabulary; and every record carries an update date and approval state. Duplicate keys are rejected rather than silently aggregated. Missing production or month prevents scenario calculation because both variables are required by the similarity model. A missing failure-family distribution can be tolerated only if the dashboard marks the profile as unavailable and does not reuse a previous chart.

Numeric validation is performed before the comparison begins. Negative production, downtime, or incident values are invalid. Efficiency must lie within the organization-defined percentage range. Failure-family shares must be nonnegative and sum to one hundred within a rounding tolerance. The history must also contain more than one production level; otherwise, planned production cannot help distinguish between months. In that case, the interface displays a warning rather than attempting a misleading comparison.

The parameters used to compare scenarios are stored with the model version. Because changing the historical records can change which months are considered most similar, an archived decision should retain the input values, data snapshot identifier, eighty-twenty priority, selected records, and output. The workbook can implement this through a protected append-only log or an exported decision record. When operating data are refreshed, regression test cases must be rerun before the new version is released.

The synthetic dataset follows the same contract. Its values were created after confidential records were removed and were manually checked for allowed ranges, unique month keys, and complete generalized failure shares. Because only qualitative ordering was preserved, the dataset is suitable for interface and calculation verification but not for estimating plant-level effect sizes.

IV. Application Architecture and Model

A. Functional Requirements

Requirements were derived from the industrial workflow and spreadsheet-risk literature. The application must allow an authorized user to select an asset group and target month, enter a planned production level, obtain indicators with one refresh action, inspect the dominant failure families, and trace each output to the records and weights that produced it. Invalid or incomplete inputs must stop calculation. Model cells must be separated from editable cells, and chart refresh must not change calculation logic.

- Usability: three controlled scenario inputs and a concise dashboard.
- Functional suitability: monthly lookup, weighted scenario estimation, indicator generation, and failure-family ranking.
- Reliability: explicit bounds, missing-data flags, and deterministic formulas.
- Maintainability: named ranges, separated worksheets, documented weights, and a version log.
- Security and confidentiality: pseudonymous identifiers, protected formula sheets, and no confidential screenshots in shared reports.

B. Layered Architecture

The workbook is organized into three logical worksheets. Input_Dashboard contains editable scenario fields and charts. History contains the monthly table and a data dictionary. Model_Audit contains comparison values, relevance scores, selected records, calculation checks, version information, and data-quality flags. This separation reduces accidental edits and makes a calculation trace visible without exposing implementation details on the user screen.

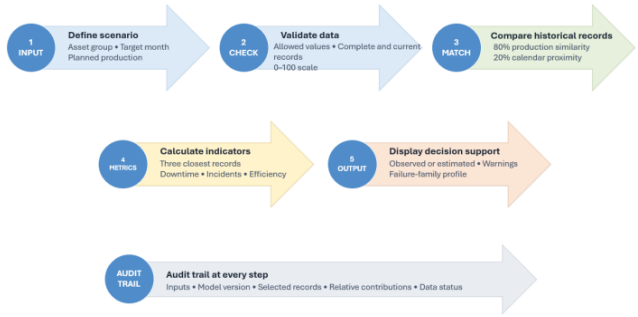


Fig. 1. Five-step information flow. Each estimate retains its inputs, selected historical records, relative contributions, and data-quality status.

As shown in Fig. 1, the application follows five visible steps: scenario entry, data checks, transparent comparison, indicator calculation, and dashboard presentation. Audit information accompanies the result and records the input, model version, three selected historical records, their relative contributions, and data-quality status.

C. Interpretable Scenario Assessment

The application uses a transparent comparison procedure rather than a black-box prediction model. For each historical month, it records the absolute difference between planned and historical production on the common 0-100 scale. It also records circular calendar distance, for which a one-month gap equals one sixth of the maximum six-month separation, and December remains adjacent to January. The two distances are combined using the selected production-calendar priority; the lowest combined value is the closest historical situation.

Decision priority: 80% production similarity + 20% calendar proximity

Production receives the larger share because workload is the main planning factor in the industrial case. Calendar proximity provides context for recurring seasonal conditions. The month comparison is circular, so December and January are treated as neighboring periods.

After comparing all historical records, the application keeps the three closest situations. Production distance is the absolute production difference divided by the full 0-100 scale. Calendar

distance is the shortest circular month gap divided by the maximum six-month separation. The two values are combined using the stated 80% production and 20% calendar priority. Records are ranked from the smallest combined distance, and smaller distances receive larger contributions by using their reciprocal after adding a numerical safeguard of 0.0001. The three contributions are then rescaled to total 100%. Three records are retained as a readability-oriented design choice rather than an optimized neighborhood size. Future validation on multi-year operational data will compare alternative record counts using out-of-sample error and result stability.

Downtime, incident, efficiency, and failure-family indicators are weighted averages of the selected records. The displayed values remain on a normalized 0-100 scale. When planned production is outside the supported historical range or no sufficiently similar case exists, the application displays a warning instead of presenting the result as certain. The pre-specified values—80% production, 20% calendar, and three records—encode industrial judgment and are configuration parameters, not fitted constants.

For transparency, the dashboard shows whether a value is observed or estimated, and the audit sheet records the historical months that contributed to the result. A maintenance planner can therefore understand the basis of an estimate without interpreting advanced mathematical notation.

This procedure is deliberately simple and auditable. It is intended for scenario assessment and maintenance planning, not for predicting the exact probability or date of a failure.

A separate exact-lookup mode retrieves descriptive statistics for a selected historical month. This dual behavior lets users distinguish recorded values from scenario estimates and prevents an estimated result from being presented as an observed fact.

D. Observed and Estimated Operating Modes

The interface uses two visually distinct operating modes. Observed mode is selected when the user requests a month that exists in the history without changing its recorded production level. The application retrieves the stored indicators and labels them as historical observations. No smoothing or load correction is applied. This mode is useful for reviewing past performance, checking source data, and discussing a completed period.

Estimated mode is selected when planned production differs from the stored level or when the user explicitly requests scenario assessment. The dashboard then labels all values as estimates, displays the contributing records, and shows how production and calendar context influenced the comparison. If no sufficiently similar historical situation exists, the application does not hide the weak match; it displays a low-similarity warning. This avoids presenting extrapolation as a well-supported estimate.

The two modes also have different audit semantics. An observed value must agree exactly with the history table and can be reconciled to a source report. An estimated value must agree with the documented step-by-step procedure and can be reconciled to the selected records and their relative contribution. Mixing these paths in a single unlabeled chart would undermine both traceability and user trust. The mode label is therefore included in every screen export.

TABLE III. ALGORITHM 1. INTERPRETABLE SCENARIO EVALUATION

Step	User-readable action
1	Confirm the asset group, target month, and planned production.
2	Compare planned production with every historical month.
3	Compare months on a circular calendar: December is next to January.
4	Apply the 80% production and 20% calendar priority; keep the three closest situations.
5	Combine their downtime, incident, efficiency, and failure-family indicators.
6	Display 0–100 results, source records, relative contributions, and warnings.

E. Spreadsheet Implementation

Data-validation lists restrict asset and month selection. Planned production accepts only numeric values within an organization-defined operating range. Named ranges replace direct sheet coordinates so that spreadsheet logic remains readable after table expansion. Exact monthly retrieval can be implemented with INDEX and MATCH or an equivalent structured lookup. For scenario assessment, the workbook ranks the historical rows, retains the closest situations, and combines their indicator values through visible worksheet calculations. The macro is limited to validation, recalculation, chart refresh, and navigation; it does not contain hidden business logic.

The interface mock-up in Fig. 2 reproduces the interaction pattern without copying any operational screen. For the worked example, the user selects a generic asset group, June, and a planned production index of 85. For May, June, and April, the normalized production distances are 0.0500, 0.1300, and 0.0800; the circular calendar distances are 0.1667, 0, and 0.3333; and the combined 80/20 distances are 0.0733, 0.1040, and 0.1307. Converting the smaller distances into larger contributions and rescaling them to 100% gives 44.1%, 31.1%, and 24.8%. The resulting downtime, incident, and efficiency indices are 41.8, 35.0, and 82.1, displayed as 42, 35, and 82. The audit sheet exposes each step of this calculation (Table III)(Fig. 3).

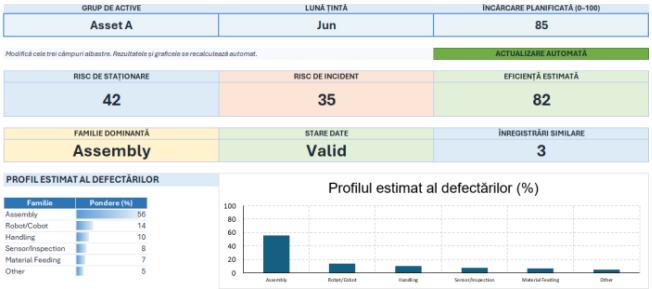


Fig. 2. Implemented Excel dashboard for the June scenario.

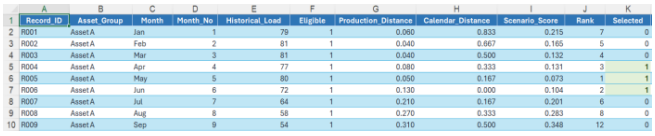


Fig. 3. Calculation data.

To reduce spreadsheet risk, editable cells use a distinct fill, formula cells are locked, and workbook structure is protected. Protection is an error-prevention mechanism rather than a security boundary. Files containing operational data must still be governed through access control, retention rules, backups, and approved distribution channels.

F. Auditability and Change Control

Calculation design follows a small number of reusable patterns. Each derived column uses one consistent calculation copied through a structured table. The eighty-twenty priority, number of reference records, accepted input ranges, and output limits are stored once in a parameter block with units and rationale. They are not repeated inside dashboard cells. This arrangement allows an auditor to verify that the same rules are applied throughout the workbook.

The audit worksheet exposes five calculation stages: validated inputs, production and month comparisons, ranked historical records, selected-reference flags, and final indicator calculations. Conditional checks confirm that the required number of references was selected, failure shares sum to one hundred, and the displayed dashboard values agree with the model outputs. A release status remains false if any check fails. The purpose is not to prove the absence of every spreadsheet defect, but to make high-impact defects observable before a decision is made.

Macro code is kept thin because hidden procedural logic is difficult for non-developers to inspect. The refresh routine reads validated inputs, requests a full recalculation, checks the release-status cell, refreshes chart series, and records a timestamp. It does not overwrite history, calculate indicators, or suppress errors. If macros are disabled, formulas remain visible, and the user can calculate manually, while the interface displays that automated refresh is unavailable.

Changes follow a lightweight release process. A maintenance-domain owner approves indicator definitions and failure families; a model owner reviews formulas and tests; an information owner approves data access and retention; and a designated user accepts the interface. The version log records the reason for change, affected ranges, reviewer, approval date, and test outcome. These controls address the principal spreadsheet risks identified in [18]-[20] without requiring a full software-development platform.

V. Verification Method

A. Test Strategy

Verification follows three levels. First, structural tests confirm that required worksheets, named ranges, validation lists, calculations, and chart sources exist. Second, numerical tests compare workbook outputs with an independent implementation of the documented calculation procedure. Third, behavioral tests exercise boundaries, invalid inputs, and changes in planned production. Software-quality criteria are aligned with functional suitability, reliability, usability, security, and maintainability from the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 25010:2023 standard [22]. Asset-management vocabulary and intent follow the International Organization for Standardization (ISO) 55000:2024 standard [23].

The verification dataset contains twelve synthetic monthly records for one generic asset group and six generalized failure families. For each month, scenarios are evaluated at ninety, one hundred, and one hundred ten percent of the reference production index, yielding thirty-six scenario calculations. Nine negative tests cover missing asset, invalid month, nonnumeric load, out-of-range load, missing indicator, stale record, failure shares not summing to one hundred, unavailable macro, and an attempted edit of a protected calculation cell (Table IV).

TABLE IV. Verification Plan and Acceptance Criteria

Verification check	Cases	What passing means
Historical replay	12	The dashboard exactly reproduces each stored month
Scenario output bounds	36	Every indicator remains between 0 and 100
Planned-production sensitivity	48 comparisons	Changing planned production changes a valid output
Invalid-input handling	9 rules	Calculation stops, and a clear warning appears
Traceability	36 scenarios	Selected records and relative contributions are visible
Confidentiality review	Entire manuscript	No company, asset identifier, product code, or raw operational value

B. Comparative Verification and Threat Controls

Comparative verification uses leave-one-month-out testing on the twelve-month synthetic sequence. One month is removed, its production level and calendar position are retained as scenario inputs, and its downtime, incident, and efficiency indices are estimated from the remaining eleven months. Three transparent baselines are used: the historical record with the nearest production level, the average of the two adjacent calendar months, and the mean of all remaining months. Performance is summarized with mean absolute error (MAE), measured in normalized index points. The same protocol is repeated with production-calendar priorities of 70/30, 80/20, and 90/10 and with two, three, or four selected records.

These tests answer whether the estimator behaves consistently and whether the pre-specified setting is competitive on the disclosed synthetic example. Because the twelve training folds overlap and come from one temporally related annual sequence, their MAEs are descriptive comparisons rather than independent estimates of generalization error. They cannot establish plant-level predictive accuracy. Historical replay verifies implementation correctness only, while parameter sensitivity verifies responsiveness rather than a causal relationship between production and failure. No maintenance-user experiment or operational impact study was conducted; these evaluations require future work with multi-year operational data.

C. Independent Oracle and Test-Case Design

Numerical verification uses an independent reference implementation rather than a duplicate set of workbook formulas. The reference implementation receives the same synthetic table and scenario inputs, repeats the documented

comparison steps in a separate execution environment, and returns the selected historical records, their relative contributions, and the resulting indicators. Workbook and reference outputs are compared at a precision appropriate to displayed values. This approach detects incorrect references, stale charts, rounding errors, and copy defects more effectively than visual inspection alone.

Test cases are partitioned by purpose. Historical replay uses the exact historical record and its recorded production level so that exact agreement is expected. Scenario cases use three reference records and three planned-load levels for every month. Boundary cases evaluate minimum, maximum, and equal production values. Negative cases remove or corrupt one required element at a time. Traceability cases compare the displayed reference list and relative contributions with the independent implementation, not only the final indicator. A failure in an intermediate field is therefore visible even when compensating errors produce a plausible total.

Acceptance criteria were defined before execution. Numerical tests require exact agreement for integer indices and a small rounding tolerance for combined results. Behavioral tests require finite outputs inside configured bounds and a changed response when planned load changes. Global monotonicity is not required because changing production can change the selected historical references; forcing monotonicity would conceal the model's actual behavior. Instead, large discontinuities are flagged for model-owner review.

VI. Results

A. Functional and Numerical Verification

The independent implementation confirmed exact historical replay for 12 of 12 months when observed mode was selected. All 36 of 36 three-level scenarios returned downtime, incident, and efficiency indices within the defined 0-100 bounds. Planned-load changes produced a finite, bounded, non-invariant risk response in 48 of 48 comparisons. All nine invalid-input cases blocked calculation rather than returning a plausible-looking value.

These results establish deterministic consistency between the documented procedure and the artifact (Table V). They do not demonstrate predictive accuracy on unseen failures. The main practical result is that a maintenance user can reconstruct why a value appeared: the selected historical records, production and month comparisons, relative contributions, and final limits are all visible.

TABLE V. Verification Results

Verification result	Outcome	Practical meaning
Historical replay	12/12 passed	Stored monthly values were reproduced correctly
Scenario bounds	36/36 passed	No result fell outside 0–100
Planned-production sensitivity	48/48 passed	The planned-production input actively influences output
Invalid-input protection	9/9 rules implemented	Invalid data cannot produce a plausible-looking result
Traceability	Complete	Every estimate shows its source records and contributions
Confidentiality	Passed	No operational identifier or raw value was disclosed

B. Sensitivity and Boundary Behavior

All forty-eight planned-load comparisons produced a finite, bounded, non-invariant response. Leave-one-month-out testing added a more demanding comparison. The proposed 80/20 configuration with three historical records achieved mean absolute errors of 9.13 for downtime, 4.34 for incidents, and 3.09 for efficiency, giving an overall mean of 5.52 index points. This was better than production-only nearest-neighbor (11.56) and the historical mean (8.06), but worse than the adjacent-calendar baseline (4.28). Fig. 4 reports all three indicators separately and prevents the overall mean from hiding where errors occur.

The baseline result shows that the disclosed synthetic sequence has a strong local calendar pattern and that the industrial 80/20 priority is a pre-specified design choice rather than an accuracy-optimal default. Boundary testing also distinguishes interpolation inside the observed production range from extrapolation outside it. The interface displays an extrapolation warning and the distance from the nearest supported load. Output limits protect the screen from impossible indices but do not make a weak scenario reliable.

Traceability tests confirmed that the dashboard, audit table, and independent implementation used the same historical references and relative contributions for every scenario. This property is central to the contribution: a reviewer can reproduce an output without inspecting macro code or accepting a black-box claim. When a result appears counterintuitive, the contributing evidence is immediately available for domain review.

C. Visibility of Seasonal and Failure Patterns

The synthetic monthly demonstration in Fig. 4 shows the exact generic sequence used for the worked example and comparative tests. Production, downtime, and incident indices share the same normalized 0-100 scale, and June is highlighted as the example target. The pattern is not evidence of causality: calendar position can represent shutdowns, staffing, product mix, environmental conditions, or maintenance campaigns.

Failure-family aggregation remains visible in the dashboard in Fig. 2. The comparative chart in Fig. 4 addresses a different question: whether the proposed estimator adds value over simple alternatives. The result is mixed rather than uniformly favorable, which supports a verification claim and motivates multi-year industrial validation before operational use.

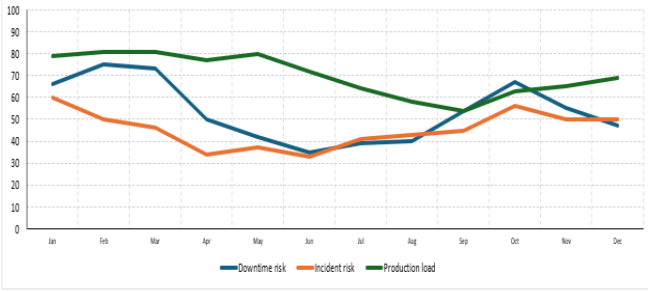


Fig. 4. Monthly evolution of the selected group.

D. Parameter Sensitivity

Table VI isolates the two configuration choices. Holding three records fixed, changing the production-calendar priority from 70/30 to 90/10 changed the overall mean absolute error from 5.45 to 6.81. Holding the 80/20 priority fixed, changing the record count from two to four changed it from 4.69 to 6.08. The mean absolute change in individual outputs relative to the pre-specified setting was 0.45 points for 70/30 with three records, 2.64 for two records, 1.57 for four records, and 2.09 for 90/10 with three records. These movements are moderate on a 0-100 scale, but the better result for two records confirms that three records are a transparency-oriented design choice, not an optimized choice. Objective selection of the record count is reserved for future validation on multi-year operational data.

TABLE VI. PARAMETER SENSITIVITY IN LEAVE-ONE-MONTH-OUT TESTING

Production/calendar priority	Records used	Downtime MAE	Incident MAE	Efficiency MAE	Overall MAE
70% / 30%	3	8.97	4.32	3.07	5.45
80% / 20%	2	7.46	4.20	2.42	4.69
80% / 20% (pre-specified)	3	9.13	4.34	3.09	5.52
80% / 20%	4	9.96	4.75	3.52	6.08
90% / 10%	3	10.78	5.88	3.77	6.81

VII. DISCUSSION

A. Answers to the Research Questions and Contribution

1) Answer to research question 1 (RQ1). The application converts three controlled inputs—generic asset group, target month, and planned production—into transparent maintenance indicators through a visible five-step workflow. Production and calendar comparisons, selected historical situations, relative contributions, outputs, and warnings remain inspectable.

2) Answer to research question 2 (RQ2). The implementation reproduced 12 of 12 historical observations, kept 36 of 36 scenario outputs inside the 0-100 bounds, produced a valid response in 48 of 48 planned-load comparisons, implemented all nine invalid-input rules, and preserved record-level traceability. These tests establish calculation consistency rather than predictive validity.

3) Answer to research question 3 (RQ3). The proposed setting outperformed production-only and historical-mean baselines but did not outperform the adjacent-calendar baseline. Its overall mean absolute error ranged from 4.24 to 6.88 across the full tested weight-neighbor grid; the isolated settings in Table VI show that the pre-specified setting is reasonably stable but not empirically optimal on one synthetic year.

4) Answer to research question 4 (RQ4). For all tested scenarios, the inputs, selected records, relative contributions, and final indicators remained visible and agreed with the independent calculation. This establishes structural traceability and reconstructability for the tested outputs. User understanding and operational suitability require future empirical study.

The contribution is not algorithmic complexity. It is the disciplined design of a transparent transition artifact. First, the model formalizes an industrial intuition—production matters more than calendar month—through visible comparisons and an eighty-twenty priority. Second, the circular treatment of the calendar avoids a common defect at the year boundary. Third, the architecture separates data, calculations, interface, and audit evidence, responding directly to spreadsheet-quality risks. Fourth, the privacy protocol demonstrates how an industrial case can be communicated without exposing performance values or asset identifiers.

The work complements predictive-maintenance literature by defining a lower-infrastructure maturity level. Sensor-based learning remains preferable when sufficient data, labels, and integration capability exist. Before that point, a low-code application can improve data definitions, reveal missing values, establish indicator ownership, and create user expectations for traceability. These capabilities reduce the organizational distance to a future CMMS, manufacturing execution, or predictive platform.

B. Practical Implications

The artifact can be piloted with limited software procurement and short training, but deployment should be conditional on local verification. Maintenance engineers can inspect the calculations, while managers receive a consistent monthly view. The baseline comparison also prevents users from assuming that additional weighting automatically improves accuracy: in a strongly seasonal process, a simple calendar rule may be the more defensible starting point.

A recommended deployment sequence is: define the data dictionary; clean at least one annual cycle; validate the workbook against independent calculations; pilot with one pseudonymous asset group; establish data and model owners; review the eighty-twenty weights; and only then extend the schema. If the application begins to influence safety-critical or high-cost decisions, migration to a governed database and tested application layer becomes necessary.

C. Governance and Migration Roadmap

Governance should scale with decision consequence. During an analytical prototype, peer review and protected formulas may be sufficient. During a controlled pilot, the organization needs approved data extracts, named owners, a versioned test pack, backup, and a restricted distribution list. Operational use requires scheduled data refresh, release approval, incident handling, training, and periodic validation. When several assets or users are added, a shared database and application service become more reliable than duplicated workbooks (Table VII).

Migration does not require discarding the artifact. The workbook can serve as an executable specification for a later system: named fields become database attributes, calculation rules become tested service functions, validation lists become reference tables, and audit fields become log events. The dashboard layout also provides a documented starting point for

a web interface. Preserving these semantics reduces the risk that a technically advanced replacement loses the interpretability expected by maintenance users.

Integration with computerized maintenance management systems (CMMS) and manufacturing execution systems (MES) should be staged. A read-only import is safer than immediate bidirectional synchronization. After identifiers and units are reconciled, the application can consume approved history extracts and export a scenario record or maintenance-planning note. Automatic work-order creation should be considered only after the decision rule, uncertainty, responsibilities, and reversal process are formally validated. Table VI summarizes the recommended maturity path.

TABLE VII. GOVERNANCE AND MIGRATION ROADMAP

Maturity stage	Required control	Evidence and readiness to progress
Prototype	Peer review of data fields and calculations	Calculation map and sample tests; data structure is stable
Controlled pilot	Named data, model, and user owners	Versioned tests and user feedback; workflow is understood
Operational workbook	Access control, release approval, and backup	Audit log and approved versions; use is repeatable and governed
Integrated application	Shared database and automated tests	Interface and security logs; several users and assets are supported
Predictive-maintenance extension	Monitored and externally validated model	Validation and model-change reports; measurable decision benefit

D. Limitations and Validity

The study has six principal limitations. The historical granularity is monthly, so short-term precursors and event sequences are unavailable. One annual cycle is insufficient to estimate stable seasonality. The pre-specified 80/20 priority and three-record setting originate from industrial judgment rather than optimization on an external dataset. Synthetic indices permit disclosure but prevent independent assessment of plant performance and may favor the calendar baseline by construction. The leave-one-month-out months are temporally related and therefore are not independent observations. Finally, no maintenance-user study was conducted, so the study does not claim improved decision time, interpretation accuracy, adoption, maintenance effectiveness, or economic benefit.

The model is also vulnerable to concept drift. Changes in product mix, automation, maintenance policy, staffing, or equipment modification can make historical similarity misleading. Data-quality flags and periodic weight review are therefore mandatory. Macros introduce compatibility and security considerations; organizations should sign approved code, disable untrusted macros, and keep business logic in visible formulas. Spreadsheet protection cannot substitute for authentication, authorization, logging, or centralized backups.

E. Future Development

Future work should validate the model on several years, asset groups, and industrial sites; compare it with multi-year seasonal-naive, regression, tree-based, and probabilistic baselines; and report uncertainty intervals. A preregistered user study should measure decision time, interpretation accuracy, perceived trust, workload, and disagreement resolution. A governed implementation can move the history to a database, connect to CMMS and MES platforms, and expose the same transparent calculations through a web interface. When condition-monitoring data become available, the scenario layer can remain as a planning view while sensor models provide short-horizon alerts.

VIII. CONCLUSION

This study presented the design and verification of an interpretable low-code application for maintenance scenario assessment in automotive assembly. The application converts controlled inputs and sparse monthly history into traceable downtime, incident, efficiency, and failure-family indicators. Its configurable comparison uses planned production and cyclic month proximity, while the selected historical records and relative contributions remain visible. Spreadsheet-quality controls, fail-closed validation, and separation of input, model, and audit layers reduce the risks of end-user computing.

The privacy-preserving case study excluded the company, specific equipment identifiers, operational capacities, absolute times, and confidential screens. Verification established calculation agreement, bounded scenario behavior, input rejection, record-level traceability, and sensitivity to parameter choices. Comparative testing showed that the proposed setting was better than production-only and historical-mean baselines but not the adjacent-calendar baseline. The artifact is therefore supported as a transparent planning aid and transition specification—not as a validated failure-probability, remaining-life, or optimization model.

DATA AVAILABILITY

The raw industrial dataset and operational workbook are not publicly available because they contain confidential manufacturing information. All values and interface elements shown in this study are synthetic, generalized, and non-identifying. The disclosed twelve-month sequence, the step-by-step worked example, baseline definitions, test counts, and acceptance criteria support reproduction of the reported verification calculations. Additional non-confidential implementation details may be provided on reasonable request, subject to institutional and publication requirements.

DECLARATION ON GENERATIVE ARTIFICIAL INTELLIGENCE

Generative artificial intelligence tools were used to assist with drafting, language refinement, calculation scripting, document organization, and visualization layout. Scientific responsibility remains with the authors, who retain full responsibility for independently verifying all domain information, calculations, results, references, interpretations, originality, and integrity of the final manuscript.

REFERENCES

[1] A. Kusiak, “Smart manufacturing,” Int. J. Prod. Res., vol. 56, nos. 1-2, pp. 508-517, 2018, doi: 10.1080/00207543.2017.1351644.

[2] J. Bokrantz, A. Skoogh, C. Berlin, T. Wuest, and J. Stahre, “Smart Maintenance: A research agenda for industrial maintenance management,” Int. J. Prod. Econ., vol. 224, Art. no. 107547, 2020, doi: 10.1016/j.ijpe.2019.107547.

[3] F. Psarommatis, G. May, and V. Azamfirei, "Envisioning maintenance 5.0: Insights from a systematic literature review of Industry 4.0 and a proposed framework," J. Manuf. Syst., vol. 68, pp. 376-399, 2023, doi: 10.1016/j.jmsy.2023.04.009.

[4] T. Zonta et al., "Predictive maintenance in the Industry 4.0: A systematic literature review," Comput. Ind. Eng., vol. 150, Art. no. 106889, 2020, doi: 10.1016/j.cie.2020.106889.

[5] P. Nunes, J. Santos, and E. Rocha, "Challenges in predictive maintenance - A review," CIRP J. Manuf. Sci. Technol., vol. 40, pp. 53-67, 2023, doi: 10.1016/j.cirpj.2022.11.004.

[6] A. Theissler, J. Perez-Velazquez, M. Kettelgerdes, and G. Elger, "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry," Reliab. Eng. Syst. Saf., vol. 215, Art. no. 107864, 2021, doi: 10.1016/j.ress.2021.107864.

[7] L. Cummins et al., "Explainable predictive maintenance: A survey of current methods, challenges and opportunities," IEEE Access, vol. 12, pp. 57574-57602, 2024, doi: 10.1109/ACCESS.2024.3391130.

[8] S. Suwatcharachaitiwong et al., "Machine Learning-Driven Preventive Maintenance for Fibreboard Production in Industry 4.0," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 3, pp. 942-950, 2025, doi: 10.14569/IJACSA.2025.0160392.

[9] M. Gopalakrishnan, M. Subramaniyan, and A. Skoogh, "Data-driven machine criticality assessment - Maintenance decision support for increased productivity," Prod. Plan. Control, vol. 33, no. 1, pp. 1-19, 2022, doi: 10.1080/09537287.2020.1817601.

[10] E. G. Gorski, E. F. R. Loures, E. A. P. Santos, R. E. Kondo, and G. R. D. N. Martins, "Towards a smart workflow in CMMS/EAM systems: An approach based on ML and MCDM," J. Ind. Inf. Integr., vol. 26, Art. no. 100278, 2022, doi: 10.1016/j.jii.2021.100278.

[11] M. Abadi, C. Abadi, A. Abadi, and H. Ben-Azza, "A Smart Decision-Making System for the Optimization of Manufacturing Systems Maintenance using Digital Twins and Ontologies," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 8, 2022, doi: 10.14569/IJACSA.2022.0130811.

[12] I. Errandonea, S. Beltran, and S. Arrizabalaga, "Digital twin for maintenance: A literature review," Comput. Ind., vol. 123, Art. no. 103316, 2020, doi: 10.1016/j.compind.2020.103316.

[13] I. Lopes et al., "Requirements specification of a computerized maintenance management system - A case study," Procedia CIRP, vol. 52, pp. 268-273, 2016, doi: 10.1016/j.procir.2016.07.047.

[14] M. P. Brundage et al., "Where do we start? Guidance for technology implementation in maintenance management for manufacturing," J. Manuf. Sci. Eng., vol. 141, no. 9, Art. no. 091005, 2019, doi: 10.1115/1.4044105.

[15] P. Muchiri, L. Pintelon, L. Gelders, and H. Martin, "Development of maintenance function performance measurement framework and indicators," Int. J. Prod. Econ., vol. 131, no. 1, pp. 295-302, 2011, doi: 10.1016/j.ijpe.2010.04.039.

[16] C. Lundgren, C. Berlin, A. Skoogh, and A. Kallstrom, "How industrial maintenance managers perceive socio-technical changes in leadership in the Industry 4.0 context," Int. J. Prod. Res., vol. 61, no. 15, pp. 5282-5301, 2023, doi: 10.1080/00207543.2022.2101031.

[17] M. Hinrichs, L. Prifti, and S. Schneegass, "Data-driven decision-making in maintenance management and coordination throughout the asset life cycle: An empirical study," J. Qual. Maint. Eng., vol. 30, no. 1, pp. 202-223, 2024, doi: 10.1108/JQME-04-2023-0038.

[18] S. G. Powell, K. R. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," Decis. Support Syst., vol. 46, no. 1, pp. 128-138, 2008, doi: 10.1016/j.dss.2008.06.001.

[19] S. G. Powell, K. R. Baker, and B. Lawson, "An auditing protocol for spreadsheet models," Inf. Manage., vol. 45, no. 5, pp. 312-320, 2008, doi: 10.1016/j.im.2008.03.004.

[20] P.-L. Poon, M. F. Lau, Y. T. Yu, and S.-F. Tang, "Spreadsheet quality assurance: A literature review," Front. Comput. Sci., vol. 18, no. 2, Art. no. 182203, 2024, doi: 10.1007/s11704-023-2384-6.

[21] Okfalisa, Mahyarni, W. Anggraini, Saktioto, and B. Pranggono, "Assessing Digital Readiness of Small and Medium Enterprises: Intelligent Dashboard Decision Support System," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 4, pp. 98-108, 2022, doi: 10.14569/IJACSA.2022.0130412.

[22] ISO/IEC 25010:2023, Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Product Quality Model, International Organization for Standardization, Geneva, Switzerland, 2023.

[23] ISO 55000:2024, Asset Management - Vocabulary, Overview and Principles, International Organization for Standardization, Geneva, Switzerland, 2024.

OptiGeoRisk: A Spatially Localized and Optimized Geo-XGBoost Framework for Credit Default Risk Classification

Claher Prastian*, Alexander Agung Santoso Gunawan
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

***Abstract*—Credit default risk prediction is important for financial institutions in geographically diverse, collateral-based lending, where early identification of risky borrowers supports preventive monitoring. Most credit scoring models treat borrower records as independent observations and rarely capture local spatial heterogeneity. This study proposes OptiGeoRisk, a spatially localized Geo-XGBoost framework for imbalanced credit default classification, using 76,742 loan transactions from West Java, Indonesia. OptiGeoRisk treats geographic coordinates as a local learning mechanism, combining haversine-based nearest-neighbor selection, adaptive neighborhood size and kernel bandwidth, and global-local probability fusion, and is evaluated against XGBoost Non-Spatial and XGBoost Spatial Feature baselines using PR-AUC, categorical and Top-K accuracy, default recall, precision, and false-positive volume. XGBoost Spatial Feature obtains the best PR-AUC, while OptiGeoRisk achieves the highest default recall, trading global ranking quality for localized early-warning sensitivity. Bootstrap resampling shows this recall advantage is statistically robust, while OptiGeoRisk's PR-AUC is statistically indistinguishable from XGBoost Non-Spatial and significantly lower than XGBoost Spatial Feature; refitting under 10 random seeds confirms this pattern is stable, and a cross-region check in DKI Jakarta shows the PR-AUC gain over a no-skill baseline holds outside West Java, though the recall advantage does not. A leakage-robust check across borrower-grouped, spatially blocked, and temporal splits shows the recall advantage holds under random and borrower-grouped partitioning but weakens when whole areas or future periods are held out, a limitation traced mainly to adaptive neighborhood selection and imbalance handling rather than the spatial kernel. These findings position OptiGeoRisk as a localized early-warning layer complementing a well-tuned global model.**

Keywords*—Credit default risk; Geo-XGBoost; spatial machine learning; class imbalance; early-warning system; financial risk management; sustainable economic growth

I. Introduction

Credit default risk prediction is increasingly important for financial institutions operating in geographically diverse markets, particularly in consumer loans, vehicle financing, pawnshops, and collateral-based lending where early identification of risky borrowers can support preventive collection strategies and portfolio monitoring. In Indonesia, financing and pawn-related activities operate under regulatory attention from the Financial Services Authority, which frames stronger risk governance, consumer protection, and responsible monitoring in this sector as part of its broader push for financial-sector stability and sustainable economic growth [1], [2].

*Corresponding author.

Credit scoring has traditionally been formulated as a statistical classification problem, and machine learning has extended this field by improving prediction from structured borrower and loan attributes [3], [4], [5]. Yet credit default modeling remains difficult because default events are typically rare compared with non-default transactions. Under this class imbalance, a model can obtain high categorical accuracy by predicting the majority class while failing to identify customers who later default [6]. For that reason, ranking-oriented and minority-class metrics such as PR-AUC and default recall are more informative than accuracy alone [7], [8].

Most credit-risk models also assume that borrower records are independent observations governed by a single global prediction function, an assumption that is restrictive for collateral-based credit in West Java, where access to branches, local economic activity, road structure, settlement density, and surrounding facilities vary substantially across regions. When location is encoded only as a category or raw coordinate, the model can learn a regional identity without explicitly adapting the prediction function to local risk behavior. That gap motivates a spatial learning framework that treats geography as a mechanism of learning rather than merely another input attribute.

This study proposes OptiGeoRisk, a spatially localized and optimized Geo-XGBoost framework for binary credit default risk classification. The framework uses real-world collateral-based loan transaction data from West Java, enriches it with branch-distance, POI, OSM, and GeoPandas-based spatial treatment, and uses coordinates structurally for local-neighborhood learning. The expected benefit is not automatic credit rejection but a practical early-warning layer that helps financial institutions prioritize monitoring, verification, and collection follow-up.

The contributions are sixfold. First, the study adapts Geo-XGBoost from spatial regression into imbalanced binary default classification. Second, it removes administrative location names and raw latitude-longitude values from model predictors while preserving coordinates for local spatial learning. Third, it evaluates the trade-off between global ranking quality and local early-warning sensitivity using metrics that are appropriate for imbalanced default risk data, benchmarked against XGBoost baselines. Fourth, it checks whether these results hold up under bootstrap resampling, ten independently retrained seeds, and a cross-region comparison between West Java and DKI Jakarta run under an identical evaluation protocol. Fifth, it isolates the

source of OptiGeoRisk's gains through leakage-robust validation across borrower-grouped, spatially blocked, and forward-chaining temporal splits, a systematic ablation of its adaptive-neighborhood, Gaussian-weighting, and fusion components, and hyperparameter-tuning budgets matched across baselines. Sixth, it reports the operational value and computational overhead of the framework at realistic review capacities, so that its deployment cost can be weighed against its early-warning benefit.

II. Related Studies

A. Credit Risk and Loan Classification

Credit scoring research provides the foundation for modeling credit default as a classification task. Classical reviews highlight the role of statistical classification in consumer credit scoring [3], while later benchmarking and systematic-review studies show that machine learning can improve predictive performance on structured credit data [4], [5]. Recent loan approval studies further demonstrate the use of ensemble learning and tree-based methods for assessing repayment feasibility and borrower creditworthiness [9], [10], [11]. More recent comprehensive reviews confirm that ensemble and gradient-boosting methods remain the strongest-performing family for structured credit data, while identifying explainability, calibration, and class-imbalance handling as the field's most persistent open challenges [12], [13].

Although these studies show that machine learning is useful for credit decision support, they generally emphasize global tabular learning. Spatial context, when available, is often treated as an ordinary input variable rather than a source of local variation in the learning process. This leaves a gap for a method that can use location to adapt prediction behavior around nearby observations.

B. Imbalanced Classification and Evaluation

The dataset used in this study has a default-class rate of only 6.87%. This confirms that the modeling problem is severely imbalanced. Previous research shows that imbalanced class distributions can affect credit-scoring model behavior and comparison [6]. PR-AUC is therefore used as the main ranking metric because Precision-Recall analysis is more informative than ROC-based interpretation when the positive class is rare [7], [8].

The study also reports categorical accuracy, Top-K categorical accuracy, default recall, precision, and false-positive volume. Top-K categorical accuracy evaluates ranked risk lists: Top-K Default checks whether actual defaults appear among the highest predicted default probabilities, while Top-K Non-default checks whether non-default cases appear among the lowest predicted default probabilities. These ranked metrics fit operational early warning better than a fixed threshold, since financial institutions usually work from prioritized customer lists rather than a single probability cutoff.

C. XGBoost, Geo-XGBoost, and Spatial Credit Modeling

XGBoost is widely used for tabular prediction because gradient-boosted trees can model non-linear relationships and interactions efficiently [14].

However, standard XGBoost is a global model. It learns one prediction function for all observations, even when the relationship between risk factors and default varies geographically.

Class-imbalance treatments such as over-sampling, under-sampling, class weighting, and SMOTE can improve minority-class learning when applied only on training data [15], [16]. Spatial learning adds another layer by allowing relationships to vary over space. Geographically weighted regression introduced the idea of estimating local relationships rather than one global function [17], [18], and spatial machine learning extends this perspective to modern predictive modeling [19]. Geo-XGBoost applies this idea to gradient-boosted trees through geographically weighted local learning [20]. Recent work extends this direction along two complementary paths: adapting XGBoost's objective directly for severe class imbalance through a generalized extreme value link and modified focal loss [21], and combining tree-boosting explicitly with spatial structure, as in spatio-temporal Gaussian-process boosting for mortgage default [22], Mixed Geographically Weighted XGBoost [23], and explainable ensemble models for peer-to-peer lending default risk [24]. These studies motivate treating geography and class imbalance as structural learning problems rather than ordinary inputs, which is the direction pursued by OptiGeoRisk.

The methodological gap addressed in this study is that Geo-XGBoost was originally proposed for regression, while credit default risk is a binary and imbalanced classification problem. OptiGeoRisk modifies the idea by using logistic probability output, imbalanced evaluation, Optuna-based optimization, adaptive local neighborhoods, and a global-local probability fusion mechanism [25].

III. Research Methodology

Fig. 2 outlines the overall research workflow followed in this study, from data preparation through model development to the spatial diagnostics presented in Section IV. The diagram groups the pipeline into three linked stages: data collection and feature construction, spatial treatment and GIS diagnostics, and the OptiGeoRisk modeling architecture and evaluation itself, which combines a global classifier with localized, geographically weighted classifiers through an adaptive fusion step before the robustness checks and cross-region validation described in Section IV. The workflow begins with the study area and dataset construction, proceeds through feature engineering and the proposed local-global fusion approach, and concludes with the diagnostic evaluation of spatial prediction performance. Each stage is described in the subsections that follow.

A. Study Area and Dataset

The study uses 76,742 collateral-based loan transactions from West Java, Indonesia, split into train, validation, and test partitions with stratified sampling. The test set contains 15,349 transactions and 1,054 default cases, a positive rate of 6.87%; this original distribution is preserved in the validation and test sets to avoid optimistic evaluation. Fig. 1 shows the geographic distribution of these transactions and the observed default rate across West Java's subdistricts (kecamatan).

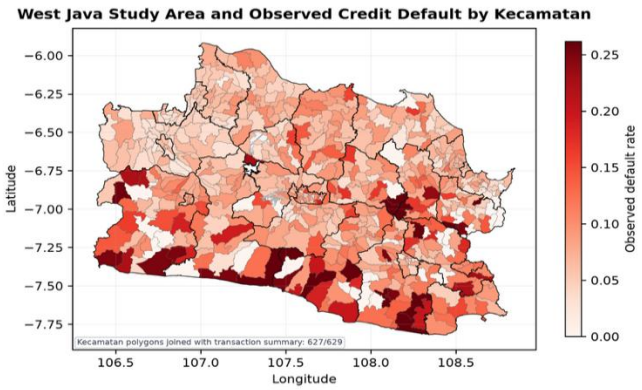


Fig. 1. West Java study area and subdistrict-level observed default distribution by Kecamatan boundary.

Spatial processing follows the style of GIS-based analytical studies. Transaction representative points were constructed from subdistrict latitude and longitude as a GeoDataFrame using CRS EPSG:4326, then reprojected when metric distance or density diagnostics were required with GeoPandas [26]. Administrative polygons were used for mapping and spatial diagnostics, not as ordinary model predictors. For detailed visualization, the kecamatan boundary layer was prepared by dissolving 5,973 village polygons into 630 kecamatan polygons.

B. Feature Engineering

The non-spatial feature group includes borrower, loan, collateral, vehicle, and temporal attributes. The spatial-feature group includes branch-distance features, POI features, and OSM-derived built-environment features. POI attributes were generated from Google Places categories within a 1,000-meter radius, including financial, healthcare, education, commerce, and market-related counts [27]. OSM features were also computed within a 1,000-meter radius, including building counts and roadway counts [28]. Coordinates are retained only for geospatial preprocessing and the local OptiGeoRisk learning mechanism.

This design reduces location-identity leakage and makes the comparison with global XGBoost more methodologically defensible. Table I reports descriptive statistics for these retained borrower, loan, collateral, and spatial-context variables.

TABLE I. DESCRIPTIVE STATISTICS OF SELECTED VARIABLES AND SPATIAL FEATURE SOURCES

Variable	Description	Mean	Std
target_macet	Default label: 1=macet, 0=lancar	0.069	0.253
loan_amount	Initial loan amount	1,961,235	927,332
collateral_value	Initial collateral value	6,054,794	2,574,454
loan_to_value	Loan-to-value	0.33	0.077
vehicle_age	Collateral age	5.533	3.054
borrower_age	Borrower age	35.041	10.854
month	Transaction month	6.824	3.298
day_of_week	Day of the week of the transaction	2.638	1.888
branch_distance	Distance from domicile to branch (km)	7.595	32.891
same_city_branch	Same city/regency as branch	0.866	0.34
poi_financial	Financial POI within 1,000 m	25.887	28.099
poi_healthcare	Healthcare POI within 1,000 m	16.526	14.615
poi_education	Education POI within 1,000 m	36.582	19.449
poi_market	Market POI within 1,000 m	7.015	7.168
osm_buildings	Building count within 1,000 m	3,498	3,318
osm_roads	Roadway count within 1,000 m	437.134	337.17
osm_build_density	Building density per km2	1,114	1,056

C. Proposed Method

OptiGeoRisk combines a global XGBoost classifier with localized, geographically weighted XGBoost models. The global component estimates P_global from the no-admin feature matrix, while the local component searches nearby training observations for each evaluation point using haversine distance, applies Gaussian kernel weighting, trains a local XGBoost classifier, and estimates P_local. The final score comes from an adaptive fusion of the two:

$$d_{ij} = \text{haversine}(g_i, g_j) \quad (1)$$

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{2b_i^2}\right) \quad (2)$$

$$P_{\text{final},i} = \alpha_i P_{\text{global},i} + (1-\alpha_i) P_{\text{local},i} \quad (3)$$

In Eq. (1), d_{ij} is the geographic distance between the prediction location and the training location. In Eq. (2), b_i is the local bandwidth derived from the neighborhood setting and w_{ij} is the resulting Gaussian kernel weight. In Eq. (3), α_i controls the observation-level global-local probability fusion between $P_{\text{global},i}$ and $P_{\text{local},i}$. Together, Eq. (1)–(3) follow the GeoXGBoost principle of combining tabular predictors X, response y, coordinates G, bandwidth, adaptive kernel weighting, and local boosted-tree learning, but the objective is changed from regression to binary logistic classification. Optuna was used in the prior optimization stage to select the imbalance strategy, XGBoost parameters, neighborhood-size baseline, kernel-bandwidth baseline, and global-local fusion parameters [25]. In the main experiment, OptiGeoRisk uses random over-sampling for the local component, initializes the local search from the optimized neighborhood and bandwidth values, and then applies adaptive neighborhood diagnostics and observation-level fusion weights during validation and testing (Algorithm 1). This integrated research framework, from feature construction through the local-global fusion mechanism to the spatial diagnostics reported in Section IV, is illustrated in Fig. 2.

Algorithm 1: Pseudocode for OptiGeoRisk Credit Default Classification

Inputs: D={X,y,G}, POI/OSM features, Optuna search space H

Output: Optimized spatial credit-risk classifier and prob_macet

Start:

Load and clean the transaction dataset

Construct loan, collateral, customer, time, branch, POI, OSM, and GeoPandas-derived features
Remove administrative-name features (kab/kota, kecamatan) to prevent leakage; retain coordinates G as the spatial mechanism
Split dataset into stratified train, validation, and test sets
For each Optuna trial t do:
Select imbalance strategy r
Sample XGBoost and spatial parameters
Apply r to training data only
Train global XGBoost classifier
For each validation coordinate g do:
Compute haversine distance to all training points
Select k nearest neighbors adaptively
Compute bandwidth b and Gaussian weights w
Train weighted local XGBoost classifier
Compute P_global and P_local
Compute adaptive alpha and P_final = alpha x P_global + (1-alpha) x P_local
Evaluate validation PR-AUC, recall, precision, and false-positive rate
Select the best trial and threshold
Evaluate final model on the test set
End

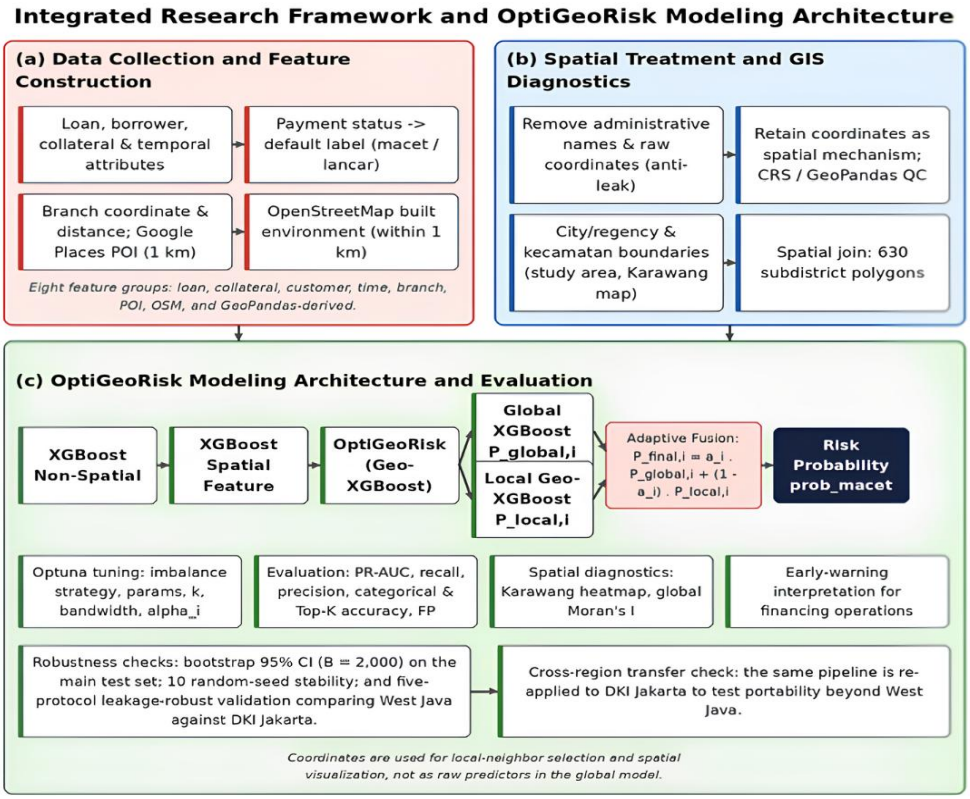


Fig. 2. Integrated research framework, feature construction, modeling architecture, and spatial diagnostic flow.

D. Spatial Diagnostics

Following spatial-data presentation used in GIS-oriented studies, the results include West Java mapping, Karawang relative risk heatmaps, feature correlation, and spatial autocorrelation diagnostics. Global Moran's I is used to evaluate whether subdistrict-level risk scores show spatial autocorrelation [29], while LISA identifies local high-high, low-low, and spatial outlier patterns [30].

E. Statistical Robustness Check

Because XGBoost Non-Spatial, XGBoost Spatial Feature, and OptiGeoRisk are compared on a single fixed train/validation/test partition, differences between point-estimate metrics could in principle reflect sampling variability in the test set rather than genuine differences between models. To assess this, non-parametric bootstrap resampling was applied to the held-out test set (15,349 transactions), following standard practice for comparing classifiers [31], [32]. For B = 2,000 resamples, test-set rows were sampled with replacement (seed = 42), and PR-AUC, default recall, precision, and categorical accuracy were recomputed for each model at its selected decision threshold. Paired differences between OptiGeoRisk and each baseline were recorded across resamples to obtain a 95% percentile confidence interval and an empirical two-sided p-value for each metric difference. Because the same resampled

indices are applied to all three models within each iteration, the comparison is paired, which removes shared sampling noise and increases the power to detect genuine differences.

This procedure quantifies evaluation uncertainty for the models as fitted; it resamples test observations, not the training process itself. Training-side variability from refitting OptiGeoRisk under multiple random seeds is assessed separately in Section IV-E, together with an initial cross-region validation check.

F. Leakage-Robust Evaluation Protocols

The main comparison in Section IV-A relies on a single stratified train, validation, and test partition, which reproduces the overall 6.87% default rate but does not by itself rule out spatial, borrower-level, or temporal leakage: a borrower with several transactions, a collateral location that repeats across partitions, or overlapping time windows could all inflate apparent performance under a purely random split. To check this, four additional partitioning protocols were applied to the same West Java dataset and the same three core models, each retrained and re-evaluated on its own 60/20/20 train, validation, and test split: borrower-grouped, in which all transactions from the same borrower (no anggota) are kept within a single partition; spatial-blocked by city/regency (kab/kota sekarang), in which entire regencies are held out for validation and testing; spatial-blocked by subdistrict (kecamatan sekarang), using a finer administrative unit; and forward-chaining temporal, in which transactions are ordered chronologically so that training data always precedes the validation and test periods. The dataset supports all four protocols: it contains 60,044 unique borrowers, of whom 12,038 have more than one transaction and are therefore a potential source of borrower-level leakage under a purely random split, 27 unique regencies and 581 unique subdistricts for the spatial-blocking protocols, and complete transaction dates from January to December 2025 for the temporal protocol.

G. Fair Hyperparameter Tuning for Baseline Models

OptiGeoRisk's Optuna-based optimization stage searches over the imbalance strategy, XGBoost parameters, neighborhood size, kernel bandwidth, and fusion weight, so an unequal tuning effort could by itself explain part of its advantage over the baseline models. To rule this out, XGBoost Non-Spatial and XGBoost Spatial Feature were each re-tuned with a standardized search space and an equal budget of 50 Optuna trials, covering the same core hyperparameters used for OptiGeoRisk (number of estimators, tree depth, learning rate, subsample and column-sample ratios, minimum child weight, gamma, and L1/L2 regularization), the same candidate imbalance strategies, and the same threshold-selection objective, maximizing default recall subject to a minimum precision of 0.11 on the validation set.

H. Systematic Ablation Study

OptiGeoRisk combines several mechanisms at once: localized neighborhood learning, adaptive neighborhood-size selection, adaptive Gaussian kernel weighting, adaptive global-local probability fusion, and imbalance handling. To isolate which of these mechanisms drives the reported gains, a six-configuration ablation matrix was built, and each configuration was retrained and evaluated on the same West Java test set. A0 is a global-only XGBoost spatial-feature baseline with no local component. A1 adds a local model with a fixed neighborhood size, fixed Gaussian weighting, and a fixed fusion weight. A2 replaces the fixed neighborhood in A1 with adaptive neighborhood and bandwidth selection, while fusion remains fixed. A3 is the full OptiGeoRisk configuration, adding adaptive global-local fusion on top of A2. A4 is a variant of A3 in which Gaussian distance weighting is replaced by uniform neighbor weights. A5 is a variant of A3 in which imbalance handling is removed from both the global and local training stages. Comparing A0 to A1 isolates the effect of adding a local component at all, A1 to A2 isolates adaptive neighborhood selection, A2 to A3 isolates adaptive fusion, A3 to A4 isolates Gaussian kernel weighting, and A3 to A5 isolates imbalance handling.

I. Operational Top-Capacity Evaluation

PR-AUC and default recall summarize model behavior across the full range of thresholds, but a collection or verification team can typically review only a limited share of the portfolio. To translate ranking quality into an operationally meaningful comparison, the three core models were additionally evaluated at fixed review capacities corresponding to the top 1%, 5%, and 10% of test-set transactions ranked by predicted default probability. For each capacity, the number of true default cases captured, precision within the reviewed set, the resulting false-positive workload, and the lift over a no-skill (random-selection) baseline were computed, together with the total default-loan exposure represented by the captured cases, to approximate the cost and benefit of prioritizing review effort by model ranking rather than by chance.

J. Computational Overhead Assessment

Localized geographically weighted learning is more expensive than a single global model because OptiGeoRisk fits a separate local classifier for each test-time neighborhood in addition to its global component. To quantify this cost, training time was recorded for the three core models on the same data split and hardware, together with OptiGeoRisk-specific overhead indicators: the number of local neighborhood models fit at test time, the mean neighborhood size (k) selected adaptively per observation, and the mean adaptive bandwidth, in kilometers, selected per observation. Training time is reported both in absolute terms and as a ratio relative to XGBoost Non-Spatial so that the additional cost of localized learning can be weighed against its early-warning benefit.

IV. Results and Discussion

A. Model Performance

Table II compares three model configurations: XGBoost Non-Spatial, XGBoost Spatial Feature, and OptiGeoRisk. Pairing the non-spatial and spatial feature sets isolates the effect of the spatial features themselves, while OptiGeoRisk adds the localized learning mechanism on top, which makes the results easier to interpret.

XGBoost Spatial Feature obtains the highest PR-AUC (0.1158) and categorical accuracy (0.7029): the strongest overall probability ranking and thresholded correctness among the compared models. Branch-distance, POI, and OSM spatial

features therefore add useful signal beyond the non-spatial credit attributes. This comparison is later corroborated under standardized hyperparameter-tuning budgets in Section IV-G.

TABLE II. EVALUATION COMPARISON ON THE MAIN TEST SET

Model	PR-AUC	Recall	Precision	Cat. Acc.	FP
XGB Non-Spatial	0.1057	0.4877	0.1025	0.6715	4,502
XGB Spatial Feature	0.1158	0.4658	0.1094	0.7029	3,997
OptiGeo Risk	0.1094	0.5256	0.1055	0.6614	4,697

OptiGeoRisk records the highest default recall (0.5256), capturing 554 actual default cases, but it also produces the largest false-positive volume (4,697). It is therefore not best interpreted as a universal replacement for global XGBoost; its main value is localized early-warning sensitivity, finding more actual default cases at the cost of a greater review workload. A bootstrap robustness check (Section III-E) confirms that this recall advantage is statistically significant relative to both baselines, while the PR-AUC gap relative to XGBoost Spatial Feature is also statistically genuine rather than sampling noise (Table III).

TABLE III. BOOTSTRAP 95% CONFIDENCE INTERVALS FOR THE MAIN TEST-SET METRICS (B = 2,000)

Model	PR-AUC (95% CI)	Recall (95% CI)
XGB Non-Spatial	0.1057 (0.0972–0.1168)	0.4877 (0.4575–0.5163)
XGB Spatial Feature	0.1158 (0.1049–0.1294)	0.4658 (0.4346–0.4949)
OptiGeoRisk	0.1094 (0.0999–0.1213)	0.5256 (0.4957–0.5542)

Paired bootstrap comparison shows that OptiGeoRisk's default recall is significantly higher than both XGB Spatial Feature ($\Delta = +0.0595$, 95% CI [0.0370, 0.0833], $p = 0.000$) and XGB Non-Spatial ($\Delta = +0.0383$, 95% CI [0.0086, 0.0678], $p = 0.010$). In contrast, OptiGeoRisk's PR-AUC is significantly lower than XGB Spatial Feature ($\Delta = -0.0066$, 95% CI [−0.0130, −0.0006], $p = 0.033$) but not significantly different from XGB Non-Spatial ($\Delta = +0.0039$, 95% CI [−0.0031, 0.0112], $p = 0.273$). The recall advantage over both baselines is therefore a robust effect rather than sampling noise, and the PR-AUC shortfall relative to XGBoost Spatial Feature is statistically genuine rather than an artifact of a single test split. As additional context, the no-skill PR-AUC baseline implied by the 6.87% test-set default prevalence is 0.0687, so all three models rank default risk substantially better than random, although the absolute PR-AUC margin above this baseline remains modest (0.037–0.047).

B. Spatial Risk Visualization and Autocorrelation

The Karawang heatmap compares observed default rate and model-based average prob_macet at the subdistrict level. The map should be read by panel: darker red indicates higher relative risk intensity within the same panel, while point size reflects the number of transactions. Observed data use actual default rates, whereas model panels show probability scores rather than final 0/1 decisions.

TABLE IV. GLOBAL MORAN'S I FOR KARAWANG RISK SURFACES

Panel	Moran's I	p-sim	z-sim
Observed	-0.0759	0.382	-0.421
XGB Non-Sp.	-0.1034	0.283	-0.626
XGB Spatial Feature	0.2977	0.004	3.028
OptiGeoRisk	0.2946	0.011	2.93

As Table IV reports, observed default data in Karawang do not show statistically significant global spatial autocorrelation in this test aggregation, whereas both the XGBoost Spatial Feature and OptiGeoRisk probability panels produce positive and statistically significant Moran's I values on the kecamatan-level visualization. The model probability surfaces therefore form clearer spatial clusters than the raw observed default rate. Fig. 3 visualizes these risk surfaces and helps compare whether predicted probabilities become darker in areas that are also darker in the observed default panel.

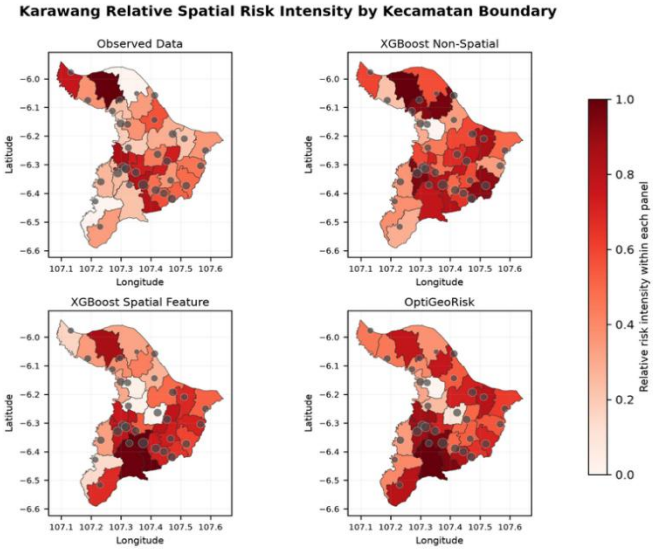


Fig. 3. Karawang relative spatial risk intensity from observed default rate and model probability scores by kecamatan boundary.

C. Feature Correlation

The strongest regional spatial-feature correlation appears after records are aggregated by city/regency, where the leading variable is OSM residential roadway count, with a Spearman-like absolute correlation of 0.6538. This supports retaining broader regional context for the main model, since the regional signal is more stable at city/regency level than at the finer kecamatan level. Fig. 4 shows this ranking: vehicle age is the only feature positively associated with default, while OSM-derived built-environment counts and same-kecamatan branch rate are the strongest negative associations.

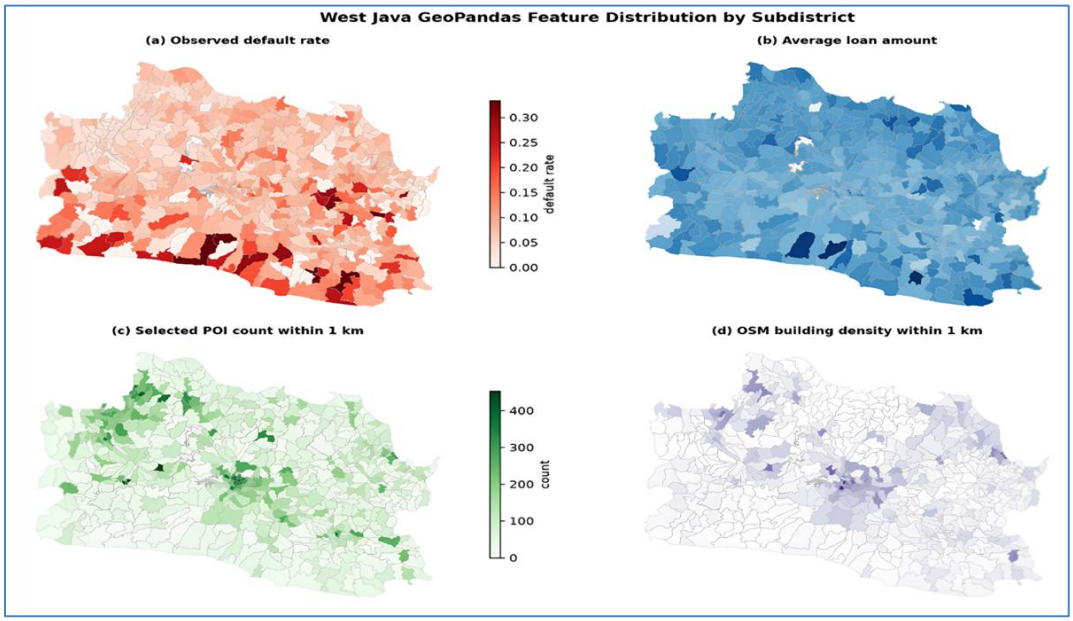


Fig. 4. West Java GeoPandas feature distribution by subdistrict: (a) observed default rate, (b) average loan amount, (c) selected POI count within 1 km, (d) OSM building density within 1 km.

The feature-correlation result is not used to claim causality; instead, it offers a transparent diagnostic that no single spatial feature dominates the target. The highly correlated feature-pair output indicates that future ablation should remove redundant density/log-count pairs before a more compact final submission model is selected.

To complement this ranking, Fig. 5 maps four of these variables across West Java at the subdistrict level: observed default rate alongside average loan amount, POI count, and OSM building density. The default-rate panel is darkest in the southern part of the study area, while POI count and building density concentrate around the central urban cluster. These two patterns occupy largely different areas, which is visually consistent with the negative correlations reported above for OSM roadway count, roadway density, and POI-related features.

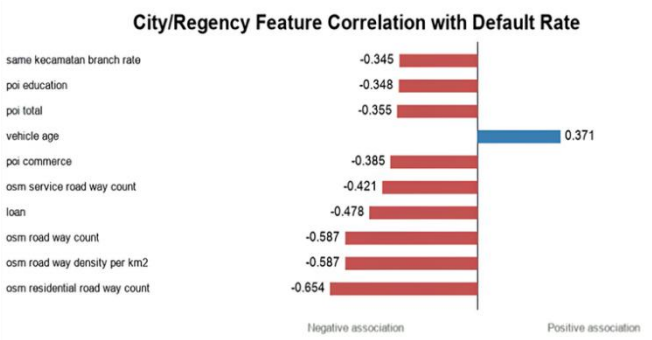


Fig. 5. City/regency-level correlation between aggregated spatial features and observed default rate.

Average loan amount does not show a comparable spatial gradient, matching its absence from the strongest correlated features in Fig. 4. As with the correlation ranking, this map is a descriptive check rather than a causal test.

D. Interpretation of the Trade-Off

The main result supports a clearer and more honest contribution statement: XGBoost Spatial Feature is preferable if the goal is the best global probability ranking, while OptiGeoRisk is more sensitive if the goal is early warning and capturing more default cases for follow-up. The model's value therefore lies in risk monitoring rather than in claiming dominance on every metric, which matters for journal review because it explains why a more expensive localized model may still be useful when a global spatial-feature baseline has stronger PR-AUC. Sections IV-F through IV-J extend this comparison to leakage-robust validation protocols, fair hyperparameter tuning, a systematic ablation of OptiGeoRisk's components, operational review-capacity analysis, and computational overhead, and refine this trade-off further rather than overturning it.

E. Seed Stability and Cross-Region Validation

To assess whether the point-estimate differences in Section IV-A generalize beyond a single train/validation/test split, all three models were refit under 10 independent random seeds on the West Java data. This procedure resamples the training process itself, which complements the bootstrap check in Section III-E (test-set resampling only) and directly addresses training-side variability.

TABLE V. STABILITY ACROSS 10 RANDOM SEEDS (WEST JAVA)

Model	PR-AUC Mean	PR-AUC Std	Recall Mean	Recall Std	FP Mean
XGB Non-Spatial	0.1076	0.0028	0.4018	0.0674	3,472
XGBoost Spatial Feature	0.1138	0.0033	0.4572	0.0640	3,931
OptiGeoRisk	0.1135	0.0034	0.4703	0.0607	4,058

As Table V reports, OptiGeoRisk keeps the highest mean default recall (0.4703) across the 10 seeds, ahead of XGBoost

Spatial Feature (0.4572) and XGBoost Non-Spatial (0.4018), with a standard deviation (0.0607) comparable to the other models. Its mean PR-AUC (0.1135) remains close to, though marginally below, XGBoost Spatial Feature (0.1138). OptiGeoRisk does not win on every individual seed, so the fairest characterization is that it is competitive and consistently more recall-sensitive rather than uniformly dominant.

As a further check on geographic portability, the same three core models were re-applied, unchanged in architecture, to a second and structurally different region, DKI Jakarta (16,292 transactions, 818 defaults), and evaluated under the same five validation protocols used for West Java in Section IV-F (transaction-random, borrower-grouped, spatial-blocked by city/regency, spatial-blocked by subdistrict, and forward-chaining temporal), so that the two regions are compared under an identical evaluation protocol rather than under two different validation schemes. West Java covers roughly 35,000 km2 with heterogeneous urban and rural sub-districts, while DKI Jakarta covers about 660 km2 of dense urban area, making it a meaningfully different geographic setting rather than a repeated sample of the same distribution. Table VI reports each model's mean PR-AUC, recall, and precision across the five protocols for both regions.

TABLE VI. IDENTICAL-PROTOCOL COMPARISON: WEST JAVA VS. DKI JAKARTA (MEAN ACROSS FIVE VALIDATION PROTOCOLS)

Region	Model	PR-AUC	Recall	Precision
West Java	XGB Non-Spatial	0.1110	0.3995	0.1119
West Java	XGB Spatial Feature	0.1116	0.4677	0.1101
West Java	OptiGeoRisk	0.1112	0.4680	0.1098
DKI Jakarta	XGB Non-Spatial	0.0795	0.1473	0.0693
DKI Jakarta	XGB Spatial Feature	0.0925	0.2185	0.0907
DKI Jakarta	OptiGeoRisk	0.0886	0.1269	0.1038

Because West Java and DKI Jakarta are now evaluated under the same five-protocol suite rather than under two different validation schemes, the comparison in Table VI is directly comparable across regions. Averaged across all five protocols, the PR-AUC ranking is consistent in both regions: XGBoost Spatial Feature obtains the highest mean PR-AUC in West Java (0.1116) and in DKI Jakarta (0.0925), with OptiGeoRisk close behind in West Java (0.1112) but comparatively further behind in DKI Jakarta (0.0886), where the training set is roughly one-fifth the size of West Java's and defaults are sparser.

Default recall does not transfer as consistently. In West Java, OptiGeoRisk's mean recall (0.4680) is essentially tied with XGBoost Spatial Feature's (0.4677), reflecting that OptiGeoRisk leads on only two of the five individual protocols (Section IV-F); in DKI Jakarta, XGBoost Spatial Feature has the clearly highest mean recall (0.2185), while OptiGeoRisk (0.1269) falls behind even XGBoost Non-Spatial (0.1473). A plausible explanation is that OptiGeoRisk's neighborhood and bandwidth settings were selected on West Java data and applied unchanged to DKI Jakarta, whose land area is roughly 1/53 that of West Java, so the same absolute bandwidth is far less local relative to the study area and fewer neighbors are available per query point when regional training data is scarce. This is reported as an honest limitation: the PR-AUC ranking is reasonably stable across these two regions, but OptiGeoRisk's recall advantage is West Java-specific under this evaluation and should not be assumed to transfer to a new region without region-specific recalibration of its neighborhood and bandwidth parameters.

F. Leakage-Robust Validation Across Evaluation Protocols

Table VII reports PR-AUC, default recall, precision, and false-positive count for the three core models under the transaction-random split used in Section IV-A and under the four additional protocols described in Section III-F. Under transaction-random and borrower-grouped partitioning, OptiGeoRisk retains the highest recall among the three models (0.5417 and 0.4961), consistent with the main result in Table II. Under the two spatially blocked protocols, however, the ranking changes: when entire regencies are held out, XGBoost Spatial Feature obtains the highest recall (0.4982), narrowly ahead of OptiGeoRisk (0.4887); when entire subdistricts are held out, XGBoost Non-Spatial is marginally ahead (0.4185) of OptiGeoRisk (0.4175). Under forward-chaining temporal validation, XGBoost Spatial Feature again leads (0.4170), with OptiGeoRisk second (0.3960) and XGBoost Non-Spatial lowest (0.2814).

TABLE VII. LEAKAGE-ROBUST EVALUATION ACROSS FIVE VALIDATION PROTOCOLS (WEST JAVA)

Protocol	Model	PR-AUC	Recall	Precision	FP
Transaction Random	XGB Non-Spatial	0.1079	0.4915	0.1027	4,526
Transaction Random	XGB Spatial Feature	0.1113	0.5332	0.1056	4,759
Transaction Random	OptiGeoRisk	0.1101	0.5417	0.1058	4,824
Borrower Grouped	XGB Non-Spatial	0.1149	0.3806	0.1143	3,013
Borrower Grouped	XGB Spatial Feature	0.1115	0.4795	0.1121	3,881
Borrower Grouped	OptiGeoRisk	0.1117	0.4961	0.1111	4,055
Spatial Blocked (City/Regency)	XGB Non-Spatial	0.1083	0.4255	0.1118	4,646
Spatial Blocked (City/Regency)	XGB Spatial Feature	0.1069	0.4982	0.1046	5,866
Spatial Blocked (City/Regency)	OptiGeoRisk	0.1068	0.4887	0.1044	5,767
Spatial Blocked (Subdistrict)	XGB Non-Spatial	0.1075	0.4185	0.1057	3,606
Spatial Blocked (Subdistrict)	XGB Spatial Feature	0.1071	0.4106	0.1058	3,533
Spatial Blocked (Subdistrict)	OptiGeoRisk	0.1072	0.4175	0.1060	3,584
Forward-Chaining Temporal	XGB Non-Spatial	0.1166	0.2814	0.1251	2,147
Forward-Chaining Temporal	XGB Spatial Feature	0.1212	0.4170	0.1225	3,258
Forward-Chaining Temporal	OptiGeoRisk	0.1201	0.3960	0.1217	3,118

These results indicate that OptiGeoRisk's recall advantage is not a fixed property of the model but depends on the validation protocol, and specifically on whether the geographic locations being scored were already represented in the training data. When borrowers or transactions are held out, but their neighborhoods

remain in the training set (transaction-random and borrower-grouped protocols), the local-learning mechanism has historical neighbors to draw on, and OptiGeoRisk's recall advantage holds. When entire geographic areas are held out (the two spatial-blocked protocols), OptiGeoRisk has no local history for the withheld locations and falls back close to, or slightly behind, the global XGBoost Spatial Feature baseline. Forward-chaining temporal validation shows a related pattern: because later transactions are more likely to fall in locations not yet densely represented in the earlier training window, OptiGeoRisk's advantage weakens there as well. The practical implication is that OptiGeoRisk should be positioned as a localized early-warning layer whose value depends on sufficient local historical coverage, rather than as a model expected to outperform a global baseline in newly opened or thinly covered territories.

G. Fair-Tuned Baseline Confirmation

Table VIII reports XGBoost Non-Spatial and XGBoost Spatial Feature under the standardized 50-trial Optuna search described in Section III-G, using the same transaction-random test split as Table II.

TABLE VIII. FAIR-TUNED BASELINE COMPARISON UNDER A STANDARDIZED 50-TRIAL SEARCH (WEST JAVA)

Model	PR-AUC	Recall	Precision	FP
XGB Non-Spatial	0.1069	0.4953	0.1040	4,499
XGB Spatial Feature	0.1108	0.5171	0.1058	4,605

Under equal tuning effort, XGBoost Spatial Feature's PR-AUC (0.1108) remains close to its originally reported value (0.1158) and continues to exceed XGBoost Non-Spatial's fair-tuned PR-AUC (0.1069), confirming that XGBoost Spatial Feature is a strong global baseline independent of tuning budget. Its default recall, however, rises from 0.4658 under the original tuning to 0.5171 under the standardized objective, which narrows its gap with OptiGeoRisk's recall (0.5256 in Table II) from 0.0598 to 0.0085. This indicates that part of OptiGeoRisk's originally reported recall advantage over XGBoost Spatial Feature reflected differences in tuning effort rather than the local-learning mechanism alone, while its PR-AUC shortfall relative to XGBoost Spatial Feature persists under fair tuning. XGBoost Non-Spatial's fair-tuned recall (0.4953) similarly improves over its original value (0.4877) but still trails OptiGeoRisk's. Read together with Section IV-D, this supports a more cautious framing of OptiGeoRisk's contribution: it remains competitive and is still the more recall-sensitive of the three core models under identical-protocol validation (Section IV-F), but its margin over a properly tuned XGBoost Spatial Feature baseline is smaller than the untuned comparison suggested, and its PR-AUC trade-off against that baseline is not resolved by tuning.

H. Systematic Ablation Results

Table IX reports PR-AUC, recall, precision, and false-positive count for the six ablation configurations described in Section III-H, all evaluated on the transaction-random West Java test set.

Adding a local component at all (A0 to A1) does not by itself improve recall: A1's recall (0.5256) is close to A0's (0.5332), and PR-AUC is essentially unchanged. Adaptive neighborhood and bandwidth selection (A1 to A2) raises recall from 0.5256 to 0.5427, the largest single gain in the matrix, confirming that letting the neighborhood adapt to local data density, rather than fixing it, is the main source of OptiGeoRisk's recall sensitivity. Adding adaptive global-local fusion (A2 to A3) trades a small amount of that recall back (0.5427 to 0.5417) in exchange for fewer false positives (4,937 to 4,824), showing that the fusion mechanism mainly helps control false-positive workload rather than raise recall further. Replacing Gaussian kernel weighting with uniform neighbor weights (A3 to A4) changes almost nothing on this test set (PR-AUC 0.1101 in both, recall 0.5417 in both, false positives 4,824 versus 4,828), so Gaussian weighting is not the mechanism responsible for OptiGeoRisk's reported gains here, at least at the neighborhood sizes selected on this dataset. Removing imbalance handling (A3 to A5) has the largest effect of all: recall falls sharply from 0.5417 to 0.4440 and false positives drop from 4,824 to 3,923, confirming that imbalance handling, not the spatial mechanism, is doing much of the work in keeping recall high. Taken together, the ablation attributes OptiGeoRisk's early-warning sensitivity mainly to adaptive neighborhood selection and imbalance handling, with adaptive fusion serving a complementary role in workload control and Gaussian weighting contributing little beyond what uniform local weighting already achieves on this split.

TABLE IX. SYSTEMATIC ABLATION OF OPTIGEORISK COMPONENTS (WEST JAVA)

Configuration	PR-AUC	Recall	Precision	FP
A0: Global spatial baseline	0.1113	0.5332	0.1056	4,759
A1: Fixed local + Gaussian + fixed fusion	0.1112	0.5256	0.1066	4,641
A2: Adaptive neighborhood + fixed fusion	0.1113	0.5427	0.1038	4,937
A3: Full OptiGeoRisk	0.1101	0.5417	0.1058	4,824
A4: No Gaussian weighting	0.1101	0.5417	0.1058	4,828
A5: No imbalance handling	0.1113	0.4440	0.1066	3,923

I. Operational Value at Review Capacity

Table X reports the number of default cases captured, recall and precision within the reviewed set, and false-positive workload for the three core models at review capacities of 1%, 5%, and 10% of the 15,349 test-set transactions.

At every capacity tested, XGBoost Spatial Feature captures the most default cases in absolute terms: 32 of 1,054 at 1% capacity, 118 at 5%, and 217 at 10%, corresponding to a lift over random selection of 3.05x, 2.24x, and 2.06x, respectively. OptiGeoRisk captures fewer cases at each capacity (27, 107, and 197) but still clearly outperforms random selection (2.57x, 2.03x, and 1.87x lift) and remains close to XGBoost Non-Spatial, which captures 20, 102, and 196 cases at the same capacities. In business terms, prioritizing the top 5% of transactions by XGBoost Spatial Feature's score captures loan exposure of approximately Rp 220.3 million in eventual defaults out of Rp 2,025.4 million total default exposure in the test set, against a false-positive workload of 649 transactions and a break-even review cost of about Rp 183,332 per false positive; OptiGeoRisk captures a similar but somewhat lower share, about Rp 199.5 million, at a workload of 660 transactions. This

operational view reinforces the finding from Sections IV-D and IV-G that XGBoost Spatial Feature is the stronger model for ranking-based prioritization at realistic review capacities, while OptiGeoRisk's comparative advantage lies less in capturing the largest absolute number of defaults at a fixed capacity and more in its recall behavior under protocols where local historical coverage is available (Section IV-F).

TABLE X. OPERATIONAL VALUE AT TOP 1%, 5%, AND 10% REVIEW CAPACITY (WEST JAVA)

Capacity	Model	Defaults Captured	Recall@ Capacity	Precision @Capacity	FP
Top 1%	XGB Non-Spatial	20	0.0190	0.1307	133
Top 1%	XGB Spatial Feature	32	0.0304	0.2092	121
Top 1%	OptiGeoRisk	27	0.0256	0.1765	126
Top 5%	XGB Non-Spatial	102	0.0968	0.1330	665
Top 5%	XGB Spatial Feature	118	0.1120	0.1538	649
Top 5%	OptiGeoRisk	107	0.1015	0.1395	660
Top 10%	XGB Non-Spatial	196	0.1860	0.1277	1,339
Top 10%	XGB Spatial Feature	217	0.2059	0.1414	1,318
Top 10%	OptiGeoRisk	197	0.1869	0.1283	1,338

J. Computational Overhead

Table XI reports training time and OptiGeoRisk-specific overhead indicators for the three core models on the same 46,044-row training split.

TABLE XI. COMPUTATIONAL OVERHEAD FOR THE THREE CORE MODELS (WEST JAVA)

Model	Fit Time (s)	Ratio vs. XGB Non-Spatial	Local Groups (Test)	Mean Bandwidth (km)
XGB Non-Spatial	2.08	1.00x	-	-
XGB Spatial Feature	12.28	5.91x	-	-
OptiGeoRisk	145.88	70.13x	609	9.6

XGBoost Spatial Feature trains in 12.28 seconds, about 5.91 times longer than XGBoost Non-Spatial's 2.08 seconds, reflecting the added cost of branch-distance, POI, and OSM feature computation. OptiGeoRisk is substantially more expensive still, at 145.88 seconds, about 70.13 times XGBoost Non-Spatial's training time, because it fits a separate local classifier for each of 609 test-time neighborhood groups, using a mean adaptively selected neighborhood size of about 52 observations and a mean adaptive bandwidth of about 9.6 kilometers. This overhead is a real deployment consideration: OptiGeoRisk's benefit is concentrated in specific validation settings (Section IV-F), and its absolute recall gain over a fairly tuned baseline is modest (Section IV-G), so its added computational cost should be weighed against how frequently a portfolio needs to be re-scored and against simpler alternatives, such as periodic rather than continuous retraining of the local component.

These fit-time figures do not by themselves show how expensive OptiGeoRisk is to score new transactions, how much of that cost is spatial neighbor search rather than local-model fitting, or how its memory footprint compares with a global classifier. To characterize these, the same transaction-random split was rescored in a separate, instrumented reproduction that isolates distance computation from local-model fitting: scoring the 15,349-row test set took 5.1 ms for XGBoost Non-Spatial and 6.5 ms for XGBoost Spatial Feature (both under 0.5 ms per 1,000 transactions), against roughly three to four thousand times that latency per transaction for OptiGeoRisk, because each of its 609 candidate kecamatan-level neighborhoods requires a freshly fit local XGBoost model rather than a single forward pass (593 of the 609 candidates contained both classes and were actually fit). Of OptiGeoRisk's total scoring time, only about 8% was spent computing haversine distances from each neighborhood centroid to every training transaction and ranking them by proximity; the remaining 92% was local-model fitting and prediction. Because this reproduction ran on different hardware than Table XI's own timings, its absolute seconds are not directly comparable to Table XI; the informative findings are the latency gap itself and the small, roughly one-tenth, share of it attributable to distance computation.

To assess how this overhead scales with the size of the historical transaction pool, the same 609 test-time neighborhoods were rescored while reducing the training pool available for neighbor search to 25%, 50%, and 75% of its original 46,044 rows. Distance-computation time grew roughly in proportion to the training-pool size, consistent with the current implementation computing a haversine distance from each neighborhood centroid to every available training transaction rather than using a spatial index such as a KD-tree or ball-tree. Local-model fit-and-predict time, in contrast, stayed essentially flat across all four pool sizes, because the adaptive neighborhood-selection rule caps each local model at 200 nearest neighbors (a mean of about 52 to 57 neighbors was actually used); growing the historical pool changes which transactions are nearest, not how many are used to fit each local model. In practice, OptiGeoRisk's per-scoring-pass latency is therefore driven mainly by the number of distinct geographic neighborhoods that must be refit, so it should grow primarily as the model's geographic footprint expands to new kecamatan, and only mildly as transaction history accumulates within kecamatan already covered; a spatial index over the training coordinates is a natural way to remove the linear distance-computation cost documented here as that history keeps growing.

Peak resident memory across the full reproduction, including loading the dataset, one-hot encoding the categorical branch, kecamatan, and vehicle-type fields into a roughly 3,400-column feature matrix, fitting both global baselines, and fitting all 593 local models needed to score the test set, stayed at approximately 530 MB when the feature matrix is kept in its native sparse form

(on the order of 40 nonzero entries per row out of about 3,400 columns). A serialized global XGBoost model occupies roughly 170 KB; a single local model occupies roughly 110 to 120 KB. Because local models are fit on demand and discarded once a neighborhood is scored, OptiGeoRisk never holds more than one local model in memory at a time, but unlike the two global baselines, whose inference-time footprint is limited to their small serialized model file, OptiGeoRisk must additionally keep the entire historical training feature matrix and coordinate set resident at serving time to support on-demand neighbor search and refitting.

V. Conclusion

This study proposed OptiGeoRisk as a spatially localized and optimized Geo-XGBoost framework for credit default risk classification. The study uses real-world collateral-based loan transactions from West Java and addresses two connected issues, severe class imbalance and geographic heterogeneity in local default patterns, by removing administrative location names and raw coordinate predictors from the ordinary feature matrix. The experiment then evaluates whether coordinates can be used more appropriately as a structural mechanism for local learning.

The main test result shows that XGBoost Spatial Feature achieves the best PR-AUC (0.1158) while OptiGeoRisk achieves the highest default recall (0.5256), a trade-off between global ranking performance and localized early-warning sensitivity. XGBoost Spatial Feature obtained the highest PR-AUC because its global learning structure produced a more stable overall ranking of default risk across the full test set, while still benefiting from branch-distance, POI, and OSM-derived spatial-context variables.

OptiGeoRisk, by contrast, used coordinates to adapt predictions through local neighborhood learning and global-local probability fusion. This localized mechanism improved sensitivity to actual default cases and produced the highest default recall, but it also introduced more false positives and slightly reduced global ranking consistency. OptiGeoRisk should therefore not be interpreted as universally superior to XGBoost Spatial Feature; rather, it is more useful as an early-warning layer when the operational priority is to capture more potential default cases for follow-up. A bootstrap significance check supports this framing: the recall gain is robust across resamples, while the PR-AUC shortfall relative to XGBoost Spatial Feature is also statistically genuine (Section III-E; Table III). A fair-tuned check (Section IV-G) shows that part of this recall margin over XGBoost Spatial Feature narrows once both baselines receive an equal hyperparameter-tuning budget, while the PR-AUC trade-off persists, so OptiGeoRisk is best read as competitive rather than categorically superior.

For financial institutions, pawnshop businesses, consumer loan providers, and collateral-based financing companies, this behavior can be valuable as an early-warning layer that ranks customers or territories needing earlier verification, preventive contact, or portfolio monitoring. It should not be used as an automatic rejection mechanism; instead, it supports human and operational decision-making, where catching risky cases earlier may be more valuable than minimizing every false alarm. This framing does not depend on the recall advantage generalizing everywhere: the PR-AUC gain over a no-skill baseline held in both regions tested (Section IV-E), so the early-warning case for OptiGeoRisk rests on the ranking gain as much as on recall. Section IV-I quantifies this concretely: at a 5% review capacity, the three core models capture between 102 and 118 of the 1,054 test-set default cases, at least 1.9 times more than the roughly 53 expected under random selection at the same capacity.

The study has several limitations and future directions. First, although Sections IV-F and IV-G now evaluate leakage-robust validation protocols and fair, equal-budget hyperparameter tuning for the three core models, the systematic ablation, computational overhead, and operational-capacity analyses in Sections IV-H through IV-J were likewise run only for these three models. Second, the cross-region check now uses an identical five-protocol evaluation for West Java and DKI Jakarta, which resolves the earlier protocol mismatch, but still covers only one additional region with a substantially smaller and sparser transaction set; broader validation across more provinces, product types, and time periods is still required. Third, the leakage-robust protocols in Section IV-F show that OptiGeoRisk's recall advantage depends on local historical coverage and weakens or reverses when whole locations or future time periods are held out, which should be treated as a deployment constraint rather than a property to be assumed; region-aware or coverage-aware bandwidth calibration is a concrete next step. Fourth, kecamatan polygons improved visualization detail but did not improve the main model when used as granular spatial features; future work should test hierarchical spatial modeling rather than simply adding finer administrative predictors. Fifth, OptiGeoRisk's computational overhead, about 70 times XGBoost Non-Spatial's training time (Section IV-J), means that frequent full retraining may not be practical at larger portfolio scale; future development should test business-cost-based thresholds, probability calibration, periodic rather than continuous retraining schedules, and model monitoring under portfolio drift.

Author Contributions

C. Prastian designed the study, processed the dataset, implemented the Geo-XGBoost and OptiGeoRisk models, and wrote the initial draft. A.A.S.G. supervised the research, reviewed the spatial autocorrelation and bootstrap analyses, and revised the manuscript. Both authors reviewed and approved the final manuscript.

Dataset Availability

The primary loan transaction dataset used in this study contains proprietary and personally identifiable borrower information from a collateral-based financing institution in West Java, Indonesia, and is therefore not publicly available due to customer privacy obligations and institutional confidentiality agreements, consistent with Indonesia's Personal Data Protection Law (UU No. 27/2022) and OJK regulations on financial-sector data governance. De-identified or aggregated data supporting the findings of this study may be made available by the corresponding author upon reasonable request and subject to a data-sharing agreement with the data-providing institution. The publicly available auxiliary datasets used for spatial feature engineering, OpenStreetMap building and road-network data [28] and administrative boundary polygons can be freely

accessed at https://www.openstreetmap.org. Point-of-interest features were derived from the Google Places API [27], accessible under Google's standard commercial API terms of service.

DECLARATION ON GENERATIVE AI

During the preparation of this manuscript, the author(s) used a generative AI tool (Claude, Anthropic) to assist with literature screening and citation verification, statistical robustness analysis (bootstrap resampling and significance testing), and language editing of the manuscript text. After using this tool, the author(s) critically reviewed, verified, and edited all AI-assisted content, and take full responsibility for the accuracy, originality, and scientific validity of the final manuscript. The core research design, experimental implementation, data analysis, interpretation of results, and conclusions are the authors' own work.

REFERENCES

[1] Otoritas Jasa Keuangan, "Peraturan Otoritas Jasa Keuangan Republik Indonesia Nomor 39 Tahun 2024 tentang Pergadaian." 2024. [Online]. Available: https://ojk.go.id/id/regulasi/Documents/Pages/POJK-39-Tahun-2024-Pergadaian/POJK%2039%20Tahun%202024%20Pergadaian.pdf

[2] Otoritas Jasa Keuangan, "Roadmap Pengembangan dan Penguatan Pergadaian 2024–2028." 2024.

[3] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: A review," J. R. Stat. Soc. Ser. A, vol. 160, no. 3, pp. 523–541, 1997, doi: 10.1111/j.1467-985X.1997.00078.x.

[4] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," Eur. J. Oper. Res., vol. 247, no. 1, pp. 124–136, 2015, doi: 10.1016/j.ejor.2015.05.030.

[5] F. Louzada, A. Ara, and G. B. Fernandes, "Classification methods applied to credit scoring: Systematic review and overall comparison," Surv. Oper. Res. Manag. Sci., vol. 21, no. 2, pp. 117–134, 2016, doi: 10.1016/j.sorms.2016.10.001.

[6] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," Expert Syst. Appl., vol. 39, no. 3, pp. 3446–3453, 2012, doi: 10.1016/j.eswa.2011.09.033.

[7] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 233–240. doi: 10.1145/1143844.1143874.

[8] T. Saito and M. Rehmsmeier, "The Precision-Recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLOS ONE, vol. 10, no. 3, p. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[9] N. Uddin, Md. K. U. Ahamed, M. A. Uddin, Md. M. Islam, Md. A. Talukder, and S. Aryal, "An ensemble machine learning based bank loan approval predictions system with a smart application," Int. J. Cogn. Comput. Eng., vol. 4, pp. 327–339, 2023, doi: 10.1016/j.ijcce.2023.09.001.

[10] A. Ananda Surya, D. R. Darmawan, and A. Solichin, "Prediksi Kapabilitas Calon Debitur Menggunakan Analisis Data Machine Learning Dengan Metode Random Forest," J. Algoritma, vol. 22, no. 1, pp. 777–788, 2025, doi: 10.33364/algoritma/v.22-1.1929.

[11] W. Andriani, Gunawan, and N. N. P. W. Naja, "Analisis perbandingan machine learning untuk prediksi kelayakan kredit perbankan pada Bank BRI Tegal," IT-Explore J. Penerapan Teknol. Inf. Dan Komun., vol. 4, no. 1, pp. 82–92, 2025, doi: 10.24246/itexplore.v4i1.2025.pp82-92.

[12] A. A. Montevechi, R. de C. Miranda, A. L. Medeiros, and J. A. B. Montevechi, "Advancing credit risk modelling with Machine Learning: A comprehensive review of the state-of-the-art," Eng. Appl. Artif. Intell., vol. 137, p. 109082, 2024, doi: 10.1016/j.engappai.2024.109082.

[13] H. Ayari, R. Guetari, and N. Kraïem, "Machine learning powered financial credit scoring: a systematic literature review," Artif. Intell. Rev., vol. 59, 2025, doi: 10.1007/s10462-025-11416-2.

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[16] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.

[17] C. Brunsdon, A. S. Fotheringham, and M. E. Charlton, "Geographically weighted regression: A method for exploring spatial nonstationarity," Geogr. Anal., vol. 28, no. 4, pp. 281–298, 1996, doi: 10.1111/j.1538-4632.1996.tb00936.x.

[18] A. S. Fotheringham, C. Brunsdon, and M. Charlton, Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Chichester, U.K.: Wiley, 2002.

[19] K. Kopczewska, "Spatial machine learning: New opportunities for regional science," Ann. Reg. Sci., vol. 68, no. 3, pp. 713–755, 2022, doi: 10.1007/s00168-021-01101-x.

[20] G. Grekousis, "Geographical-XGBoost: A new ensemble model for spatially local regression based on gradient-boosted trees," J. Geogr. Syst., vol. 27, no. 2, pp. 169–195, 2025, doi: 10.1007/s10109-025-00465-4.

[21] J. Mushava and M. Murray, "A novel XGBoost extension for credit scoring class-imbalanced data combining a generalized extreme value link and a modified focal loss function," Expert Syst. Appl., vol. 202, p. 117233, 2022, doi: 10.1016/j.eswa.2022.117233.

[22] P. Kündig and F. Sigrist, "A spatio-temporal machine learning model for mortgage credit risk: Default probabilities and loan portfolios," Eur. J. Oper. Res., vol. 331, no. 3, pp. 1022–1041, 2026, doi: 10.1016/j.ejor.2025.10.039.

[23] F. Gao, S. Y. He, and M.-P. Kwan, "Mixed Geographically Weighted XGBoost (M-GWXGB) model: A new spatially explicit machine learning model," Ann. Am. Assoc. Geogr., vol. 116, pp. 824–855, 2025, doi: 10.1080/24694452.2025.2586039.

[24] M. Atef, S. Ouf, W. Seoud, and M. I. Gabr, "A novel approach using explainable prediction of default risk in peer-to-peer lending based on machine learning models," Neural Comput. Appl., vol. 37, pp. 21783–21803, 2025, doi: 10.1007/s00521-025-11489-8.

[25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.

[26] K. Jordahl and others, "GeoPandas: Python tools for geographic data." 2020. doi: 10.5281/zenodo.3946761.

[27] Google, "Places API." [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/overview

[28] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org." 2026. [Online]. Available: https://www.openstreetmap.org

[29] P. A. P. Moran, "Notes on continuous stochastic phenomena," Biometrika, vol. 37, no. 1/2, pp. 17–23, 1950.

[30] L. Anselin, "Local indicators of spatial association-LISA," Geogr. Anal., vol. 27, no. 2, pp. 93–115, 1995, doi: 10.1111/j.1538-4632.1995.tb00338.x.

[31] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap. New York, NY, USA: Chapman & Hall/CRC, 1994. doi: 10.1201/9780429246593.

[32] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," Neural Comput., vol. 10, no. 7, pp. 1895–1923, 1998, doi: 10.1162/089976698300017197.

Collaborative Development Method for Improving Uncertainty Management in Agile Software Development

Maha Makkass[1], Youness Laghouaouta[2], Adil Anwar[3]
Siweb, EMI, Mohammed V University in Rabat, Rabat, Morocco[1, 3]
National Institute of Posts and Telecommunications (INPT), Rabat, Morocco[2]

***Abstract*—Software development projects frequently face uncertainty arising from evolving requirements, communication gaps, and inconsistent coordination among stakeholders. This study presents the Collaborative Development Method (CDM), an agile framework designed to strengthen collaboration, improve requirements engineering practices, and support uncertainty management throughout the software development lifecycle. CDM integrates collaborative activities, requirements traceability, user story validation, and continuous alignment between business objectives and technical implementation within a structured development process. To evaluate its effectiveness, an empirical study was conducted with software developers from Moroccan software companies. The measurement model was first validated using confirmatory factor analysis, and the proposed relationships were then examined through a Robust Least Squares regression model after diagnostic tests indicated violations of ordinary least squares assumptions. The results show that collaboration among project stakeholders, structured communication, requirements traceability, user story validation, and business–technical alignment significantly improve perceived uncertainty management. In contrast, the use of a Domain-Specific Language for structuring user stories and adapting iteration length to functionality size did not demonstrate a statistically significant effect. These findings provide empirical evidence that CDM offers a promising framework for improving coordination, requirements management, and decision-making in agile software development environments.**

***Keywords*—Agile leadership; uncertainty management; agile software development; requirements engineering; Collaborative Development Method (CDM); software process improvement**

I. Introduction

The rapid evolution of digital projects has profoundly transformed the conditions of design, coordination, and management within development teams. In this context, organizations are confronted with environments characterized by the variability of needs, the frequency of functional adjustments, the diversity of the actors involved, and the uncertainty that accompanies the translation of business expectations into technical solutions. Faced with these constraints, Agile Software Development has emerged as a privileged framework for improving team responsiveness and strengthening their adaptive capacity. However, the relevance of this approach does not lie solely in its iterative rhythm or in the application of practices derived from Scrum. It also depends on the way collaboration among stakeholders is organized, on how requirements are clarified and monitored, and on the ability of agile leadership to support continuous mechanisms of dialogue, validation, and adjustment. From this perspective, uncertainty management constitutes a central issue, since project performance depends on the ability to reduce ambiguities without rigidifying the development process. The question, therefore, is not only whether agility makes it possible to respond more quickly to change, but also under what conditions it helps make such change more understandable, better coordinated, and more fully integrated into the overall course of the project, particularly through the integration of a collaborative agile DSL.

It is within this logic that the present study focuses on the role of the application of CDM in improving uncertainty management in a development environment. The analysis is based on the idea that the control of uncertainty depends on the quality of interactions among actors, on the structuring of requirements, and on the alignment between business objectives and technical choices. It therefore lies at the intersection of Agile Software Development, Requirements Engineering, and Software Process Improvement. Within this framework, the use of formalization mechanisms, particularly through a collaborative agile Domain-Specific Language (DSL), may be regarded as a support intended to make needs more explicit, strengthen the consistency of user stories, and facilitate exchanges among the different project stakeholders. However, the actual scope of these instruments depends on their integration into a collective dynamic sustained by agile leadership capable of articulating coordination, learning, and adaptation. The objective of this research is therefore to examine the extent to which several dimensions related to the application of CDM contribute to improving uncertainty management in software projects. Through this approach, the study seeks to better understand how agile practices can move beyond their procedural dimension to become a concrete lever for the stabilization, intelligibility, and sustainable improvement of the development process.

II. Literature Review

A. Agile Management and Uncertainty Management

Agile management can be presented as a mode of governance particularly well suited to software development when uncertainty affects needs, technical choices, and the conditions under which the project is carried out. Highsmith

[19] recalls that the deliverables of a project cannot be fully specified before its start and must emerge throughout its implementation, which from the outset places software development within a logic of adjustment. From the same perspective, Highsmith and Cockburn [20] show that high iteration frequencies strengthen teams' ability to cope with risk and uncertainty, particularly through the continuous integration of customer feedback. Although agile practices can implicitly contribute to risk reduction through iterative development and continuous feedback, explicit risk management mechanisms remain important for systematically identifying, assessing, and monitoring project risks [36]. Schwaber and Sutherland [40] describe Scrum as a lightweight framework aimed at generating value through adaptive solutions to complex problems, relying on empiricism, Lean Thinking, transparency, inspection, and adaptation. Waterman [47] also explains that agility, risk, and uncertainty are closely linked from the design phase onward. In the same vein, Hazzan and Dubinsky [17] present agility as a philosophy that makes it possible to deal with uncertain situations and processes of change. Agile management thus appears, in software development, as a project management framework that organizes adaptation rather than assuming the initial stability of development conditions.

This logic becomes particularly important when software development takes place in environments marked by technological and organizational turbulence. Ahmed et al. [1] emphasize that volatile, uncertain, complex, ambiguous, radical, and rapid contexts intensify the challenges associated with change management, thereby increasing the relevance of agile management. Akkaya and Ahmed [2] extend this idea by showing that the transition to Industry 5.0 reinforces the need for rapid adaptation through computing power capable of handling complex problems more effectively. Conforto et al. [7] indeed observe that agile project management is attracting increasing interest beyond software development precisely because it responds to the frequent evolution of requirements. Zaslavska and Zaslavskyi [49], for their part, consider that the agile project-based approach ensures the flexibility and adaptability of structures in a turbulent environment. Lill et al. [27] likewise argue that agility proposes flexibility as a response to uncertainty, whereas methods based on detailed planning are better suited to relatively stable situations. Finally, Teece et al. [43] relate this orientation to organizational agility, showing that it makes it possible to dynamically adapt the firm's capabilities in the face of risk and uncertainty in the innovation economy.

In the field of software development, agile management is not limited to absorbing external uncertainty; it also transforms the way teams interpret and address it. Fagerholm and Pagels [13] thus show that a register of uncertainty reduction appears in lean and agile values through the intention to base actions on evidence rather than on unjustified orders. Dönmez and Grote [11] indicate that agile software development teams perceive uncertainty both as a threat and as an opportunity, which directly influences their project management practices. Salin and Lundgren [39] add that agile risk management is based on the acceptance of uncertainty as a normal component of the organizational environment, as well as on the creation of a climate of psychological safety that makes it possible to discuss openly errors, failures, and emerging uncertainties.

Levitt [26], for his part, emphasizes the delegation of authority to competent actors and the existence of shared global awareness, elements that strengthen software teams' ability to react rapidly to the unexpected.

The role of agile management appears particularly clearly in Requirements Engineering, where uncertainty directly concerns the understanding, prioritization, and evolution of needs. Logue and McDaid [29] show that agile approaches can address the uncertainty inherent in software development projects through a requirements prioritization and planning method that incorporates variability management. Their use of statistical simulation is intended to model fluctuations in requirements and estimates so as not to freeze specifications prematurely. Highsmith and Cockburn [20], for their part, emphasize that the continuous integration of customer feedback through closely spaced iterations allows the product to be adjusted more precisely to changing requirements. Ries [38] advocates an approach based on continuous innovation, iterative development, and the regular validation of hypotheses with customers, rather than on a rigid plan. Mendonça de Sá Araújo et al. [33] also show that Design Thinking practices, such as collaborative ideation, empathy, problem framing, divergent thinking, and rapid prototyping, can be structured with agile frameworks to better capture business opportunities in an uncertain environment. In software development, Requirements Engineering thus becomes an evolutionary process of clarifying needs rather than a stage of definitive requirements fixation.

This logic of continuous adjustment extends into Software Process Improvement, where improving the development process depends less on fixed standardization than on a capacity for the progressive revision of practices. Schwaber and Sutherland [40] explain that Scrum makes the relative effectiveness of existing practices visible so as to enable improvements, thereby giving agility an explicit process-improvement function. Tavares et al. [42] show that the analysis of risk management in Scrum projects makes it possible to integrate uncertainty management into the very core of agile practices. Van Oorschot et al. [46] nevertheless point out that iteration length influences the uncertainty of outcomes and that cycles that are too short or too long may produce fluctuations that are unfavorable to performance. Kreye et al. [24] confirm that the detailed planning of project means becomes of limited relevance as long as deliverables remain uncertain, which justifies significant replanning between sprints. Cooper [8] adds that the combination of agile methods and stage-gate can offer flexibility, speed, and improved communication in new product development. From a similar perspective, Lishner and Shtub [28] show that risk and uncertainty must be integrated into the very choice of development methodology, which reinforces the strategic scope of agility in improving the software process.

Agile management also affects the quality of software development through the forms of coordination and learning that it makes possible among team members. Gren et al. [15] show that teamwork, open communication, and agile planning

are strongly correlated with perceived agility in large companies, which highlights the importance of collaborative mechanisms in software environments. Tolfo et al. [45] explore the relationship between agile practices and the development of entrepreneurial skills in software development, suggesting that these practices can strengthen capabilities that are useful in evolving environments. Hendler [18] shows that, when it comes to articulating physical and digital development, close collaboration, Scrum ceremonies, and increment planning facilitate mutual adjustment and continuous learning, which is decisive in phases of high uncertainty. Azonuche and Enyejo [4], for their part, emphasize that agile risk management is based on the decentralization of responsibility, combined with the use of real-time data and predictive dashboards enabling continuous adaptation. Dobrowolski et al. [10] add that agile methodology implies a constant readiness to initiate change, to learn from each evolution, and to create value for the customer. In software development, this combination of collaboration, learning, and distributed decision-making strengthens teams' ability to maintain project continuity despite uncertainty.

B. CDM as a New Method for Agile Uncertainty Management

The Collaborative Development Method can be justified as a structured extension of the principles of Agile Software Development, but also as an operational framework of Agile Management oriented toward the control of uncertainty [30,31]. Fig. 1 indeed shows that the method does not rely solely on a succession of activities, but on a precise organization of exchanges among the Customer, the Product Owner, and the development team. From operation 1, in which the client explains business needs, the method places the clarification of business needs at the center, which is consistent with Alsaqqa et al. [3], for whom agile development is characterized by its ability to integrate changing requirements throughout the project. This interpretation is reinforced by Rasheed et al. [37], who recall that agility gives priority to interactions, collaboration with the customer, and rapid response to change, as well as by Perkusich et al. [35], who present agility as a change-driven approach based on the empirical learning of customer needs. Through operation 2, in which the Product Owner creates the User Story and updates the priorities, and then operation 3, in which the PO validates the new US and prioritized feature, Fig. 1 highlights a logic of framing, arbitration, and continuous validation that falls directly within Agile Management. This logic corresponds to the view of Gheorghe et al. [14], who define Agile Software Development as a set of values and principles oriented toward adaptive, incremental, and iterative production, carried out by self-organized teams. It also aligns with Kuhrmann et al. [25], who emphasize that agility is as much a mindset and a culture as it is a mere process. Accordingly, CDM can be understood as a form of agile management aimed at better structuring decisions, stabilizing exchanges, and, above all, preventing the emergence of ambiguities that fuel uncertainty in a development context.

The specific value of the method becomes even more apparent when one considers the way Fig. 1 organizes the transformation of need into understandable and shareable requirements. After the first validation, operation 4 indicates that the PO updates his Product Backlog; then operation 5 shows that the PO presents the highest-priority US for the next Sprint, before operation 6, in which the Dev team validates the comprehension. This sequence reflects a form of Agile Management based on progressive clarification, synchronization among actors, and the reduction of interpretive gaps between business need and technical translation. This is precisely what Chen et al. [5] support with ScrumOntoSPL, by proposing a collaborative method combining Scrum, agile engineering, and uniform requirements specification based on an ontology to reduce coordination costs. The presence in Fig. 1 of the User Story respecting DSL shows that this structuring relies on a Domain-Specific Language, which strengthens the stability of formulations, the traceability of requirements, and their shared understanding. CDM thus fits within a Requirements Engineering perspective in which formalization is not opposed to agility, but rather placed at the service of a more controlled management of interactions. This interpretation is consistent with Khan and Keung [21], who show that the improvement of practices depends on an understanding of the enabling factors and barriers that influence process evolution, as well as with Choraś et al. [6], who emphasize the importance of concrete metrics to assess and guide agile practices. Although CDM shares with ScrumOntoSPL the objective of strengthening requirements coordination within an agile setting, their structural emphasis differs. ScrumOntoSPL primarily combines Scrum with ontology-based requirements specification for software product-line reuse, whereas CDM organizes the broader collaboration cycle among the Customer, Product Owner, and development team around need clarification, prioritization, traceability, validation, prototyping, and iterative adjustment. The distinctive contribution of CDM therefore lies less in introducing an isolated requirements representation mechanism than in integrating these mechanisms within a continuous uncertainty-management process.

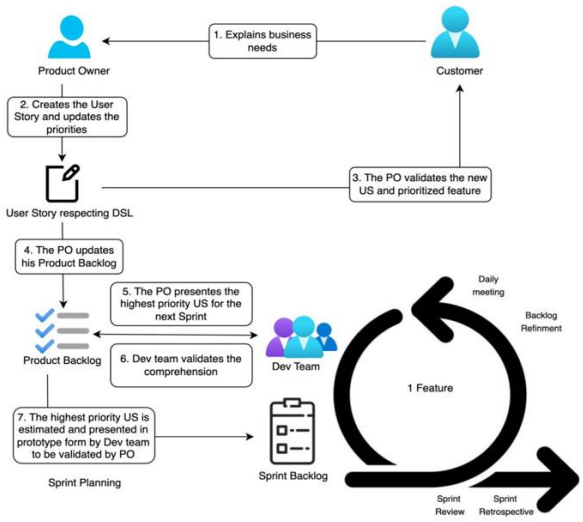


Fig. 1. Application of CDM and agile uncertainty management.

It also aligns with Marinho et al. [32], Klünder et al. [23], and Tell et al. [44], who show that, in complex or distributed environments, effective coordination requires methodological combinations capable of adjusting processes to the real constraints of collective work. Fig. 1 therefore suggests that

CDM is not only intended to manage the flow of user stories more effectively, but also to reduce upstream the sources of interpretive instability that feed project uncertainty.

The link with Uncertainty Management becomes even more explicit in the final part of Fig. 1. Operation 7 indicates that the highest priority US is estimated and presented in prototype form by the Dev team to be validated by the PO before its inclusion in the Sprint Backlog and its processing within a loop comprising Daily meeting, Backlog Refinement, Sprint Review, and Sprint Retrospective, with an iteration centered on 1 Feature. This architecture shows that CDM seeks to control uncertainty through the decomposition of needs, incremental validation, continuous feedback, and a focus on a specific functionality. In other words, Agile Management here takes the form of a sequential governance of uncertainty, in which each operation in Fig. 1 contributes to reducing part of the ambiguities likely to affect understanding, estimation, coordination, or decision-making. This interpretation is directly consistent with Gupta et al. [16], who consider risk and uncertainty management to be a critical factor in software product management. It also aligns with Ezghari and Zahi [12], who recall that estimations are affected by data inaccuracies and the limits of human reasoning; Nahar et al. [34], who emphasize the usefulness of mitigation mechanisms in the face of uncertainty; Dey and Lee [9], who show the insufficiency of requirements analysis and modeling techniques to address it properly; and Kläs and Vollmer [22], who identify specific sources of uncertainty related to scope, data quality, and model adequacy. In the same vein, Włodarski et al. [48] show that the agile approach improves team cohesion and productivity, while Sinha and Das [41] observe that it becomes particularly effective when sprints are devoted to clearly delimited functional improvements, which corresponds to the logic of 1 Feature highlighted in Fig. 1. Thus, CDM can be understood as a mechanism that articulates Collaborative Development Method, Agile Software Development, Agile Management, Scrum, Requirements Engineering, DSL, Software Process Improvement, and Uncertainty Management within the same logic of progressive development stabilization: the more needs are clarified, validated, prioritized, and tested through the operations; the more uncertainty becomes governable.

III. METHODS

A. Research Hypotheses

In light of the preceding theoretical developments, the improvement of uncertainty management in a software development environment can be understood as the result of a set of complementary mechanisms relating simultaneously to Agile Management, Requirements Engineering, and Software Process Improvement. Indeed, the literature reviewed shows that uncertainty does not stem solely from the project's external instability, but also from interpretive gaps among actors, ambiguities surrounding requirements, coordination difficulties, shortcomings in validation, and the potential mismatch between business logic and technical translation. From this perspective, the application of the Collaborative Development Method appears as a structured framework that makes it possible to better organize interactions among the client, the Product Owner, and the development team, to strengthen the progressive clarification of needs, to improve the formalization and traceability of requirements, to support their continuous validation, and to ensure a more coherent treatment of functionalities throughout the iterations. Thus, CDM cannot be reduced to a simple operational sequence, but constitutes a mechanism of collective regulation through which exchanges, priorities, formulations, and adjustments become more intelligible and more manageable. It is on this conceptual basis that the following hypotheses are formulated:

- H1: Improving collaboration among the client, the Product Owner, and the development team has a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H2: The formalization of exchanges among stakeholders and the clarification of needs from the definition phase onward have a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H3: The use of DSL to structure the formulation of user stories has a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H4: Improving requirements traceability has a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H5: The validation of user stories and functionalities with the relevant actors has a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H6: The alignment between business objectives and technical objectives has a positive effect on the perceived improvement of uncertainty management through the application of CDM.
- H7: The adaptation of the iteration to the size of the requested functionality has a positive effect on the perceived improvement of uncertainty management through the application of CDM.

B. The Model

In order to examine empirically the factors associated with the improvement of uncertainty management in a development environment, this study adopts a statistical model linking the perception of this improvement to several dimensions derived from the application of CDM. The model specified below makes it possible to assess the role of collaboration simultaneously, the formalization of exchanges, the structuring of user stories, traceability, validation, business–technical alignment, and iterative adjustment in explaining the phenomenon under study.

$$UNCDM = \beta 0 + \beta 1.COLLB + \beta 2.REQFM + \beta 3.DSLST + \beta 4.REQTR + \beta 5.VALUS + \beta 6.GOAAL + \beta 7.ITERF + \varepsilon$$

The dependent variable UNCDM refers to the perceived improvement of uncertainty management through the application of CDM. It is measured based on six items: UNCDM1: the application of CDM helps reduce ambiguities in the definition of needs; UNCDM2: the application of CDM improves the understanding of client expectations; UNCDM3: the application of CDM facilitates the anticipation of changes related to requirements; UNCDM4: the application of CDM contributes to reducing uncertainties during development; UNCDM5: the application of CDM improves the control of requirements throughout the project; UNCDM6: the application of CDM makes the development process more predictable in the face of change. This variable thus makes it possible to assess the overall contribution of CDM to a better control of uncertainty in development. The variable COLLB measures the improvement of collaboration among the client, the Product Owner, and the development team. It is based on six items: COLLB1: CDM improves communication among the client, the Product Owner, and the development team; COLLB2: CDM fosters better information sharing among stakeholders; COLLB3: CDM facilitates coordination among the actors involved in the project; COLLB4: CDM reduces misunderstandings among stakeholders; COLLB5: CDM strengthens the involvement of actors in exchanges related to requirements; COLLB6: CDM improves the quality of interactions among project members. This variable aims to capture the extent to which the application of CDM strengthens the collective dynamic surrounding requirements management. The variable REQFM corresponds to the formalization of exchanges among stakeholders and the clarification of needs from the definition phase onward. It is measured through six items: REQFM1: CDM makes it possible to formalize exchanges among stakeholders from the beginning of the project; REQFM2: CDM improves precision in the expression of needs; REQFM3: CDM helps clarify the client's functional expectations; REQFM4: CDM reduces ambiguities in the initial formulation of requirements; REQFM5: CDM facilitates the establishment of a shared understanding of needs; REQFM6: CDM improves the structuring of discussions around requirements. This variable therefore makes it possible to assess the capacity of CDM to make the initial process of needs definition more explicit and more widely shared.

The variable DSLST refers to the use of DSL to structure the formulation of user stories. It is approached through six items: DSLST1: the use of DSL makes the formulation of user stories clearer; DSLST2: the use of DSL improves consistency in the writing of user stories; DSLST3: the use of DSL reduces errors in the interpretation of user stories; DSLST4: the use of DSL facilitates the development team's understanding of user stories; DSLST5: the use of DSL makes it possible to standardize the writing of user stories; DSLST6: the use of DSL improves the overall quality of the formulation of user stories. This variable thus measures the contribution of DSL to a better structuring of the language used in the formalization of requirements. The variable REQTR refers to the improvement of requirements traceability. It is based on six items: REQTR1: CDM facilitates the monitoring of requirements from their definition to their implementation; REQTR2: CDM makes it possible to better link requirements to the functionalities developed; REQTR3: CDM improves the identification of the origin of expressed needs; REQTR4: CDM facilitates the tracking of changes made to requirements; REQTR5: CDM improves visibility regarding the evolution of requirements throughout the project; REQTR6: CDM strengthens continuity between initial needs and final deliverables. This variable makes it possible to assess the extent to which CDM strengthens informational continuity and the monitoring of needs over time. The variable VALUS measures the validation of user stories and functionalities with the relevant actors. It is assessed through six items: VALUS1: CDM facilitates the validation of user stories with the relevant stakeholders; VALUS2: CDM improves the verification of the conformity of functionalities with expressed needs; VALUS3: CDM makes it possible to obtain faster feedback on the proposed user stories; VALUS4: CDM promotes better acceptance of functionalities by the relevant actors; VALUS5: CDM reduces the risks of discrepancy between expectations and the functionalities developed; VALUS6: CDM improves the process of validating needs before final development. This variable therefore aims to capture the role of CDM in securing the process of validating requirements and functionalities.

The variable GOAAL corresponds to the alignment between business objectives and technical objectives. It is measured by six items: GOAAL1: CDM improves consistency between business objectives and technical choices; GOAAL2: CDM facilitates the translation of business needs into technical requirements; GOAAL3: CDM strengthens the alignment between client priorities and the work of the technical team; GOAAL4: CDM reduces the gaps between functional expectations and the proposed technical solutions; GOAAL5: CDM enables better articulation between business vision and technical development; GOAAL6: CDM improves the consideration of business objectives in technical decisions. This variable therefore makes it possible to assess the capacity of CDM to bring the functional and technical dimensions of the project closer together. The variable ITERF refers to the adaptation of the iteration to the size of the requested functionality. It may be measured based on six items: ITERF1: CDM makes it possible to adjust the duration of the iteration to the complexity of the functionality; ITERF2: CDM facilitates the breakdown of functionalities into suitable iterations; ITERF3: CDM improves the fit between workload and the content of the iteration; ITERF4: CDM reduces the gaps between the size of the functionality and iteration planning; ITERF5: CDM promotes better priority management in the iterative breakdown; ITERF6: CDM improves consistency between iteration planning and functional needs. This variable makes it possible to apprehend the extent to which the application of CDM makes the iterative pace better adjusted to the concrete characteristics of the functionalities to be developed. The items associated with all the variables in the model were measured using a five-point Likert scale, ranging from 1 "strongly disagree" to 5 "strongly agree."

C. Construction of Variables

After the initial construction of the variables, a confirmatory factor analysis was conducted in order to verify the convergent validity of the items and to refine the constructs. Only the items whose factor loadings were greater than 0.5 were retained, while the others were removed.

Convergent validity and construct reliability were assessed using standardized factor loadings, average variance extracted (AVE), and composite reliability (CR). AVE was calculated as $AVE = \frac{\sum_{i=1}^{n} \lambda_i^2}{n}$, where λi denotes the standardized factor loading of item i. Composite reliability was calculated as $CR = \frac{(\sum_{i=1}^{n} \lambda_i)^2}{(\sum_{i=1}^{n} \lambda_i)^2 + \sum_{i=1}^{n}(1-\lambda_i^2)}$. Items with standardized loadings below 0.50 were excluded before the final construct scores were computed. The recalculated composite reliability coefficients range from 0.787 for DSLST to 0.913 for ITERF, indicating satisfactory construct reliability. AVE values range from 0.556 for DSLST to 0.727 for ITERF and therefore exceed the conventional 0.50 threshold for all constructs. These results support convergent validity for the retained measurement model (Table I).

TABLE I. RESULTS OF THE CONFIRMATORY FACTOR ANALYSIS FOR CONVERGENT VALIDITY AND CONSTRUCT RELIABILITY

Items	Standardized Factor Loading	Composite Reliability (CR)	AVE
UNCDM 2	0.753617	0.804706	0.583555
UNCDM 3	0.889353		
UNCDM 5	0.625922		
COLLB 1	0.620869	0.858672	0.607233
COLLB 3	0.883678		
COLLB 4	0.735819		
COLLB 6	0.849197		
REQFM 2	0.960432	0.790954	0.568807
REQFM 3	0.612614		
REQFM 5	0.639292		
DSLST 1	0.879949	0.786674	0.556081
DSLST 4	0.693511		
DSLST 5	0.642631		
REQTR 2	0.736849	0.836306	0.630673
REQTR 3	0.835277		
REQTR 5	0.807084		
VALUS 2	0.720613	0.827529	0.616331
VALUS 4	0.781365		
VALUS 6	0.848044		
GOAAL 1	0.624159	0.811529	0.595338
GOAAL 3	0.744317		
GOAAL 5	0.917840		
ITERF 1	0.924147	0.912991	0.727204
ITERF 4	0.660281		
ITERF 5	0.898398		
ITERF 6	0.900931		

The examination of composite reliability and AVE further supports the psychometric quality of the retained constructs. Composite reliability values range from 0.786674 to 0.912991, exceeding the commonly accepted threshold of 0.70. AVE values range from 0.556081 to 0.727204 and remain above the 0.50 threshold for all constructs, supporting convergent validity. These results indicate that the retained items provide satisfactory reliability and explain an adequate proportion of variance in their respective constructs.

The retained constructs therefore demonstrate satisfactory convergent validity and composite reliability. In addition, the very low VIF values reported in the subsequent regression diagnostics indicate limited linear redundancy among the explanatory variables. Dedicated discriminant-validity criteria such as the Fornell–Larcker criterion or HTMT were not available from the retained analytical outputs and should therefore be examined in future replications to further consolidate the measurement-model assessment.

D. Data

Participants were software developers working in an organizational context in Morocco in which agile development practices were used. The survey specifically targeted developers involved in software development activities and exposed to collaborative requirements-management practices or to the application of CDM. Eligibility was therefore aligned with the purpose of the study, which required respondents to have sufficient practical exposure to agile development and collaborative requirements activities to evaluate the proposed mechanisms. A criterion-based purposive sampling approach was used to target software developers with relevant exposure to agile development and collaborative requirements-management practices. This sampling strategy was consistent with the study objective, as evaluation of the CDM-related dimensions required respondents with practical knowledge of the development context under investigation. The questionnaire also collected general participant characteristics, including age group, gender, software development experience, team size, use of agile methods, and exposure to CDM. Before completing the CDM-related items, respondents were provided with a presentation of the study, the conceptual role of CDM, and the main dimensions examined in the questionnaire. Participation was voluntary, anonymous, and confidential, and all constructs were assessed using five-point Likert-scale items ranging from strongly disagree to strongly agree.

Table II summarizes questionnaire completion and validity. A total of 250 questionnaires were completed by developers working in software development companies in Morocco. After data-quality screening, 154 questionnaires were retained for analysis, while 96 were excluded because they did not meet the criteria for statistical usability. Valid questionnaires therefore represented 61.6% of the completed questionnaires. Because the total number of individuals initially invited to participate was not available in the retained study records, an overall survey response rate could not be calculated.

TABLE II. QUESTIONNAIRE COMPLETION AND VALIDITY

Completed	Valid	Invalid	Valid questionnaires / Completed questionnaires
250	154	96	61.6 %

Table III, presenting the descriptive statistics, first shows a strong proximity between the means and medians for all variables, which indicates a distribution that is globally centered around the intermediate level of the scale. The means

range from 2.943182 for COLLB to 3.099567 for REQTR, while all medians are equal to 3.000000. This configuration suggests that the developers surveyed in software development companies in Morocco express moderate to slightly favorable perceptions regarding the contribution of CDM. The standard deviations, ranging from 0.723932 to 0.870004, reflect a contained dispersion, indicating some variability but without excessive heterogeneity among the responses observed in the sample.

TABLE III. DESCRIPTIVE STATISTICS

	Mean	Median	Max	Min	Std. Dev.	Skewness	Kurtosis
UNCDM	3.021645	3.000000	5.000000	1.000000	0.806361	-0.223477	2.665365
COLLB	2.943182	3.000000	5.000000	1.000000	0.772549	0.096656	2.595066
REQFM	3.010823	3.000000	5.000000	1.000000	0.846556	-0.077309	2.286297
DSLST	2.978355	3.000000	5.000000	1.000000	0.789984	0.009580	2.580690
REQTR	3.099567	3.000000	5.000000	1.000000	0.870004	0.011773	2.405846
VALUS	3.012987	3.000000	5.000000	1.000000	0.772515	-0.142265	2.598047
GOAAL	2.997835	3.000000	5.000000	1.000000	0.803493	-0.034496	2.953082
ITERF	3.040584	3.000000	5.000000	1.000000	0.723932	0.113296	2.613382

The examination of the shape indicators confirms this relative stability of the distributions. The skewness coefficients remain low, ranging from -0.223477 for UNCDM to 0.113296 for ITERF, which indicates distributions close to symmetry, with slight inclinations depending on the variable. Likewise, the kurtosis coefficients vary from 2.286297 to 2.953082, values close to the reference threshold, which suggests the absence of extreme concentrations or abnormally long tails. The minimum and maximum values cover almost the entire response scale, from 1 to 5, showing that all response categories were used by the respondents, without excessive polarization in the judgments expressed.

E. Choice of Methodology

The examination of the Ramsey RESET test (Table IV), applied to the model estimated by ordinary least squares, reveals a deficiency in statistical specification. This test aims to assess the validity of the retained functional form by detecting possible omissions of relevant variables, nonlinearity effects, or, more generally, an inadequacy between the theoretical structure of the model and the relationship actually observed in the data. In this case, the results obtained show a t-statistic of 3.009505 associated with a probability of 0.0031, an F-statistic of 9.057119 also significant at the 1% level, as well as a likelihood ratio of 9.330813 with a probability of 0.0023. The convergence of these three indicators toward the same conclusion leads to the rejection of the null hypothesis of correct specification.

The examination of the Jarque-Bera normality test applied to the residuals of the model estimated by ordinary least squares reveals a significant deviation from the normality assumption. Indeed, in Fig. 2, the Jarque-Bera statistic reaches 25.54575, with a probability of 0.000003, a value well below the conventional 5% threshold. This result therefore leads to the rejection of the null hypothesis according to which the residuals follow a normal distribution. This conclusion is consistent with the descriptive characteristics of the residual series, marked by positive skewness (skewness = 0.854400) and a kurtosis coefficient slightly above the normal reference value of 3 (kurtosis = 4.030127), reflecting a distribution that is both asymmetric and more concentrated around the mean, with relatively heavy tails.

TABLE IV. RAMSEY RESET TEST FOR MODEL SPECIFICATION (OLS)

Ramsey RESET Test Equation: UNTITLED Specification: UNCDM C COLLB REQFM DSLST REQTR VALUS GOAAL ITERF			
Omitted Variables: Squares of fitted values			
	Value	**Df**	**Probability**
t-statistic	3.009505	145	0.0031
F-statistic	9.057119	(1, 145)	0.0031
Likelihood ratio	9.330813	1	0.0023
F-test summary:			
	Sum of Sq.	**Df**	**Mean Squares**
Test SSR	0.537285	1	0.537285
Restricted SSR	9.138960	146	0.062596
Unrestricted SSR	8.601675	145	0.059322

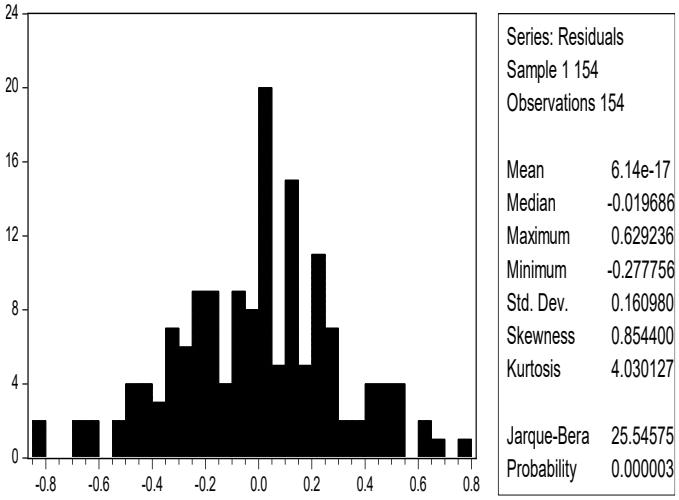


Fig. 2. Jarque-Bera normality test (OLS).

The examination of the Breusch-Pagan-Godfrey heteroskedasticity test applied to the model estimated by ordinary least squares (Table V) reveals the presence of non-constant residual variance. Indeed, the three reported statistics converge toward the same conclusion. The F-statistic stands at 2.592992 with a probability of 0.0150, the Obs*R-squared statistic reaches 17.02851 with a probability of 0.0172, while the Scaled explained SS statistic rises to 23.18845 with a probability of 0.0016. Since all these probabilities are below the conventional 5% threshold, the null hypothesis of homoskedasticity is rejected. This result means that the dispersion of the errors is not stable from one observation to another, reflecting heterogeneity in the variance of the disturbances.

TABLE V. BREUSCH-PAGAN-GODFREY HETEROSKEDASTICITY TEST (OLS)

Test statistic	Value	Degrees of freedom	Probability
F-statistic	2.592992	F(7, 146)	0.0150
Obs*R-squared	17.02851	Chi-square(7)	0.0172
Scaled explained SS	23.18845	Chi-square(7)	0.0016

In light of the diagnostics carried out on the model estimated by ordinary least squares, the use of robust regression through RLS appears methodologically justified. Indeed, the Ramsey RESET test revealed a specification deficiency, suggesting that the relationship between the variables is not fully captured by the retained linear functional form. In addition, the Jarque-Bera test led to the rejection of the normality assumption for the residuals, revealing an asymmetric and leptokurtic distribution of the errors, while the Breusch-Pagan-Godfrey test confirmed the presence of heteroskedasticity, that is, a non-constant variance of the disturbances. The combination of these three results indicates that the classical conditions ensuring the efficiency and inferential validity of OLS estimators are not fully satisfied. In this context, estimation by RLS constitutes an appropriate alternative, insofar as it makes it possible to reduce the influence of atypical observations, improve the robustness of the estimated coefficients, and produce more reliable inference when the standard assumptions of OLS are weakened.

IV. RESULTS

A. Robustness Analysis

The application of the Ramsey RESET test to the model estimated by the Robust Least Squares method indicates, this time, an improvement in statistical specification. Indeed, this test makes it possible to verify whether the retained functional form remains adequate after correcting the shortcomings identified under OLS estimation. The results in Table VI show a t-statistic of 1.149077 associated with a probability of 0.2524, an F-statistic of 1.320377 with a probability likewise equal to 0.2524, as well as a likelihood ratio of 1.395985 corresponding to a probability of 0.2374. Since all these probabilities are well above the conventional 5% threshold, the null hypothesis of correct model specification cannot be rejected. This result suggests that, under robust estimation, the explanatory model reflects the relationship under study more satisfactorily. Under robust estimation, the corresponding RESET diagnostic no longer indicated evidence against the retained functional specification.

TABLE VI. RAMSEY RESET TEST FOR MODEL SPECIFICATION (RLS)

Ramsey RESET Test Equation: UNTITLED Specification: UNCDM C COLLB REQFM DSLST REQTR VALUS GOAAL ITERF			
Omitted Variables: Squares of fitted values			
Test	Value	Df	Probability
t-statistic	1.149077	145	0.2524
F-statistic	1.320377	(1, 145)	0.2524
Likelihood ratio	1.395985	1	0.2374
F-test summary			
Test	**Sum of Sq.**	**Df**	**Mean Squares**
Test SSR	2.983366	1	2.983366
Restricted SSR	330.6080	146	2.264438
Unrestricted SSR	327.6246	145	2.259480

The analysis of the variance inflation factors applied to the model estimated by the Robust Least Squares method, presented in Table VII, shows the absence of a multicollinearity problem among the retained explanatory variables. This diagnostic is based primarily on the Centered VIF values, which make it possible to assess the intensity of the linear correlations between each regressor and all the other variables in the model. In this case, all the observed values are very low and remain well below the critical thresholds commonly accepted in the statistical literature. They range from 1.015819 for COLLB to 1.053895 for VALUS, while the other variables, namely REQFM, DSLST, REQTR, GOAAL, and ITERF, also display levels very close to unity. Such a configuration indicates that each explanatory variable provides distinct information, without significant linear redundancy with the others.

TABLE VII. VARIANCE INFLATION FACTOR ANALYSIS (RLS)

Variance Inflation Factors Sample: 1 154 Included observations: 154			
Variable	**Coefficient Variance**	**Uncentered VIF**	**Centered VIF**
C	0.313689	21.33338	NA
COLLB	0.176150	3.562348	1.015819
REQFM	0.194895	4.433069	1.049236
DSLST	0.201126	4.507918	1.045842
REQTR	0.179376	4.424994	1.024247
VALUS	0.175523	3.927746	1.053895
GOAAL	0.178327	3.655873	1.021817
ITERF	0.191091	4.874647	1.044690

The examination of the Jarque-Bera normality test applied to the residuals of the model estimated by the Robust Least Squares method in Fig. 3 indicates that the normality hypothesis cannot be rejected. Indeed, the Jarque-Bera statistic stands at 1.540661, associated with a probability of 0.462860, a

value well above the conventional threshold of 5%. This result therefore leads to the conclusion that the distribution of the residuals is compatible with a normal distribution. This conclusion is consistent with the observed descriptive characteristics, notably a very moderate skewness (skewness = 0.149395) and a kurtosis coefficient of 2.611634, which is close to the reference value associated with normality. The residual histogram also confirms a relatively balanced distribution around the mean, without marked distortion or excessive concentration in the tails of the distribution. Under these conditions, robust estimation appears to have substantially improved the distributional behavior of the errors.

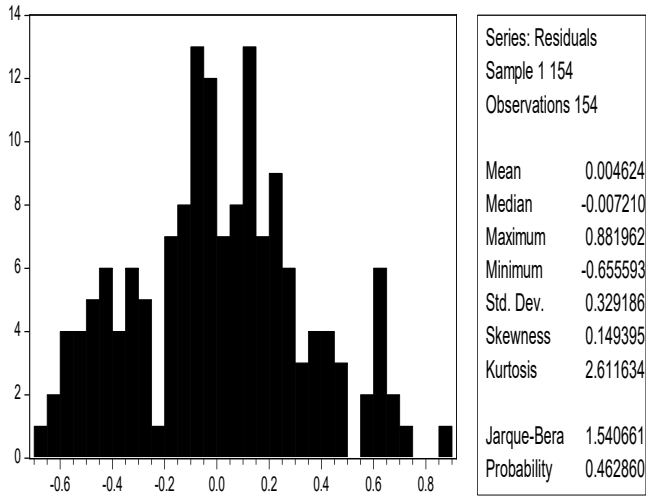


Fig. 3. Jarque-Bera normality test (RLS).

The examination of the Breusch-Pagan-Godfrey heteroskedasticity test applied to the model estimated by the Robust Least Squares method, presented in Table VIII, indicates the absence of a significant problem of residual heteroskedasticity. Indeed, the three test statistics converge toward the same conclusion. The F-statistic is equal to 0.926060 with a probability of 0.4883, the Obs*R-squared statistic stands at 6.546938 with a probability of 0.4775, while the Scaled explained SS statistic reaches 2.096089 with a probability of 0.9543. Since all these probabilities are well above the conventional 5% threshold, the null hypothesis of homoskedasticity cannot be rejected. This result means that the variance of the residuals now appears stable from one observation to another, which suggests a substantial improvement in the statistical quality of the model after the use of robust estimation. The corresponding residual diagnostics under RLS no longer indicated the irregularities observed in the initial OLS specification.

TABLE VIII. BREUSCH-PAGAN-GODFREY HETEROSKEDASTICITY TEST (RLS)

Test statistic	Value	Degrees of freedom	Probability
F-statistic	0.926060	F(7, 146)	0.4883
Obs*R-squared	6.546938	Chi-square(7)	0.4775
Scaled explained SS	2.096089	Chi-square(7)	0.9543

The examination of the influence statistics based on Covratio highlights an overall satisfactory stability of the estimation. Fig. 4 indeed shows that the Covratio values remain concentrated within a relatively narrow interval around unity, without abrupt breaks or excessive deviations likely to indicate the existence of observations exerting a disproportionate influence on the variance-covariance matrix of the coefficients. The observed fluctuations remain moderate and follow a fairly regular overall pattern, which suggests that the hypothetical removal of one observation would not substantially modify the overall precision of the estimators. In other words, the structure of the model does not appear to depend on a few atypical points capable of significantly altering the stability of the results.

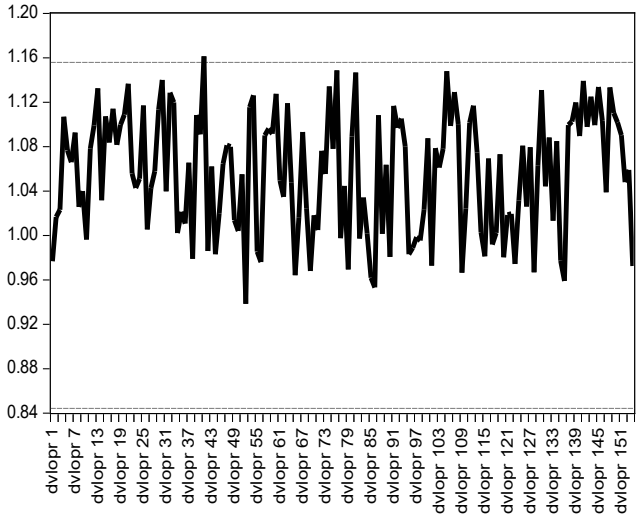


Fig. 4. Covratio influence statistics (RLS).

The initial OLS estimation revealed several statistical weaknesses through the Ramsey RESET test, the Jarque-Bera normality test, and the Breusch-Pagan-Godfrey heteroskedasticity test, indicating respectively a specification deficiency, non-normality of the residuals, and a non-constant error variance. The use of RLS estimation made it possible to substantially improve the statistical quality of the model. The results obtained after re-estimation show, through the same tests, a specification that has become acceptable, a residual distribution compatible with normality, and a more stable error structure. Finally, the examination of the Covratio influence statistics confirms the overall stability of the model.

B. Regression Results

Table IX presents the results of the estimation using the Robust Least Squares method. The estimation is based on M-estimation with Bisquare weighting, a scale founded on median-centered MAD, and Huber-type robust standard errors, which strengthens the model's resistance to disturbances likely to affect the precision of the results.

Standardized coefficients were also computed to facilitate comparison of the relative magnitude of the predictors. Among the statistically significant predictors, collaboration shows the largest standardized association with perceived uncertainty management ($\beta = 0.278$), followed by requirements formalization ($\beta = 0.238$), user-story validation ($\beta = 0.205$), requirements traceability ($\beta = 0.192$), and business–technical alignment ($\beta = 0.186$).

TABLE IX. ROBUST LEAST SQUARES ESTIMATION RESULTS

Dependent Variable: UNCDM Method: Robust Least Squares Sample: 1 154 Included observations: 154 Method: M-estimation M settings: weight=Bisquare, tuning=4.685, scale=MAD (median centered) Huber Type I Standard Errors & Covariance					
Variable	**Coefficient**	**Std. Error**	**z-Statistic**	**Prob.**	**Standardized β**
C	2.531772	0.684337	3.699599	***0.0003	
COLLB	0.289812	0.088981	3.257014	***0.0014	0.278
REQFM	0.226525	0.080578	2.811250	***0.0056	0.238
DSLST	0.001834	0.089058	0.020589	0.9836	0.002
REQTR	0.177941	0.080146	2.220210	**0.0279	0.192
VALUS	0.213618	0.090198	2.368323	**0.0191	0.205
GOAAL	0.186551	0.085948	2.170514	**0.0315	0.186
ITERF	0.122503	0.094912	1.290706	0.1987	0.110

*** Significant at 1%; ** Significant at 5%; * Significant at 10%.

COLLB is associated with a positive coefficient that is significant at the 1% level (p-value = 0.0014). This result shows that better collaboration among the client, the Product Owner, and the development team is associated with a perceived improvement in uncertainty management within the CDM framework. Hypothesis H1 is therefore accepted. This finding highlights that the quality of exchanges among project actors constitutes an important support for limiting misunderstandings and better framing adjustments throughout the development process. REQFM also displays a positive sign and significance at the 1% level (p-value = 0.0056). This result confirms that the formalization of exchanges among stakeholders and the clarification of needs from the outset contribute to a better perceived management of uncertainty. Hypothesis H2 is therefore accepted. This result suggests that a more explicit initial framing of expectations reduces margins of ambiguity and promotes more controlled project management. DSLST shows a positive coefficient, but it is not significant, even at the 10% level (p-value = 0.9836). Under these conditions, no statistically established effect can be retained between the use of DSL to structure the formulation of user stories and the perceived improvement of uncertainty management. Hypothesis H3 is therefore rejected. This suggests that the structuring of the language of user stories, by itself, is not sufficient to produce a perceptible change in the control of uncertainty.

REQTR emerges with a positive coefficient that is significant at the 5% level (p-value = 0.0279). This result indicates that better requirements traceability is associated with a perceived improvement in uncertainty management through the application of CDM. Hypothesis H4 is therefore accepted. It can be inferred that more rigorous monitoring of requirements, from their initial expression to their evolution, helps preserve project coherence and reduce zones of uncertainty. VALUS presents a positive coefficient that is significant at the 5% level (p-value = 0.0191). This result shows that the validation of user stories and functionalities with the relevant actors contributes positively to the perceived improvement of uncertainty management. Hypothesis H5 is thus accepted. This result underlines the value of shared validation in bringing developed functionalities closer to the expectations actually expressed by stakeholders.

GOAAL also displays a positive coefficient that is significant at the 5% level (p-value = 0.0315). This result confirms that alignment between business objectives and technical objectives promotes a perceived improvement in uncertainty management. Hypothesis H6 is therefore accepted. This shows that better consistency between the project's functional logic and its technical translation contributes to making development more intelligible and more stable. ITERF presents a positive coefficient, but it is not significant (p-value = 0.1987). This result therefore does not allow validation of the existence of a statistically significant effect of adapting the iteration to the size of the requested functionality on the perceived improvement of uncertainty management. Hypothesis H7 is rejected. This indicates that the adjustment of the iterative pace, considered in isolation, does not appear sufficient to strengthen uncertainty control in a clearly observable manner.

V. DISCUSSION

The results first show that the perceived improvement of uncertainty management relies to a large extent on the quality of collaboration among the client, the Product Owner, and the development team. This result confirms the idea that agility does not produce its effects solely through iteration, but also through the concrete organization of interactions among stakeholders. It is therefore consistent with Highsmith and Cockburn [20], who associate adaptive capacity with the continuous integration of feedback, as well as with Schwaber and Sutherland [40], for whom Scrum is based on transparency, inspection, and adaptation in dealing with complex problems. It also aligns with Gren et al. [15], who show the importance of teamwork, open communication, and agile planning in perceived agility. In the same logic, the positive effect of the formalization of exchanges and the initial clarification of needs confirms that the reduction of uncertainty depends on the early explication of expectations rather than on the late regulation of discrepancies. This finding extends the work of Logue and McDaid [29], who emphasize that uncertainty related to requirements can be better contained when prioritization and planning avoid freezing specifications prematurely, as well as that of Ries [38], for whom the regular validation of hypotheses makes it possible to adjust the product more precisely. The results therefore indicate that, within the CDM framework, uncertainty control depends first on the ability of the actors to construct a language of exchange that is sufficiently clear to reduce ambiguities during development.

The results then show that uncertainty management is strengthened when requirements remain visible, monitored, and validated throughout the project. The positive effect of traceability confirms that uncertainty does not arise solely from the variability of demands, but also from possible breaks between the expressed need, its intermediate reformulation,

and the actual development. This result is consistent with the logic of Software Process Improvement highlighted by Schwaber and Sutherland [40], according to which practices must be made visible in order to be improved, as well as with Tavares et al. [42], who show that risk management can be integrated into Scrum practices themselves. Likewise, the positive effect of validating user stories and functionalities is consistent with Gupta et al. [16], who consider risk and uncertainty management to be a critical factor in software projects, as well as with Dey and Lee [9], who emphasize the limitations of approaches that are insufficiently equipped to deal properly with uncertain requirements. The positive effect of alignment between business objectives and technical objectives also extends the logic described in Fig. 1, where clarification, prioritization, estimation, prototyping, and validation follow one another in order to make uncertainty progressively governable. From this perspective, the results confirm that the main contribution of CDM lies in its ability to maintain continuous coherence between functional logic and technical translation, rather than in execution speed alone.

By contrast, the results qualify the scope of certain instruments when they are considered in isolation. The absence of a statistically established effect of DSL suggests that the formal structuring of user stories is not, by itself, sufficient to produce a perceptible improvement in uncertainty management. This result does not necessarily contradict the expected contributions of a structured language, but it shows that its effectiveness remains conditional on the way it is embedded in collective coordination practices. From this perspective, the results invite a more cautious reading of the promises associated with formalization tools and are consistent with the idea defended by Kuhrmann et al. [25], according to which agility is as much a matter of mindset and culture as of process. Similarly, the absence of a significant effect of adapting the iteration to the size of the functionality shows that the iterative pace, taken in isolation, does not constitute a sufficient lever for uncertainty control. This observation is consistent with Van Oorschot et al. [46], who show that iteration length influences outcomes without automatically guaranteeing greater stability, as well as with Lishner and Shtub [28], who recall that risk and uncertainty must be integrated into the broader choice of methodology. Overall, the results therefore confirm that uncertainty management depends less on a particular technical device than on a coherent combination of collaboration, clarification, traceability, validation, and alignment, supported by agile leadership capable of articulating these dimensions within the CDM framework.

VI. Conclusions and Future Work

This study shows that improving uncertainty management in software projects depends less on the declared adoption of an agile approach than on the way its principles are effectively translated into the organization of work. The results highlight that the reduction of ambiguities becomes more tangible when actors build a shared understanding of needs, sustain continuous exchanges, and maintain close coordination throughout development. From this perspective, Agile Software Development cannot be reduced to a succession of iterations or to a simple sequence of ceremonies inspired by Scrum. Its effectiveness rests primarily on the quality of collaboration, on the circulation of information, and on the capacity of agile leadership to create a framework for dialogue, adjustment, and collective regulation. Uncertainty management thus appears as a socially constructed process, in which uncertainties are contained not through rigid control, but through frequent interactions, shared validation, and a better articulation between client expectations and development choices. Within this interpretation, CDM may be understood not merely as a procedural tool, but as a new form of agile management, insofar as it structures the relationships among the Customer, the Product Owner, and the development team around progressive clarification, prioritization, validation, and iterative adjustment. In that sense, CDM extends agile management by providing a more explicit operational architecture through which coordination becomes more readable and uncertainty more governable. The findings therefore suggest that agility produces its strongest effects when it reinforces collective coherence around requirements, so that changes become more intelligible, more openly discussed, and more easily integrated into the trajectory of the project.

The analysis also shows that the most robust contribution to project stabilization comes from practices that strengthen Requirements Engineering within an agile framework. When needs are clarified, monitored, and validated continuously, uncertainty control becomes more credible and the project gains in coherence between business logic and technical translation. In a Scrum-inspired environment, the value of the process lies less in the speed of iteration itself than in the quality of the collective trade-offs it makes possible. In this respect, the use of DSL, particularly in the form of a collaborative agile DSL, does not appear sufficient on its own to significantly transform uncertainty management, which recalls that formalization tools replace neither dialogue nor shared interpretation. The essential contribution of the study therefore concerns Software Process Improvement, understood as the joint strengthening of communication, validation, alignment, and collective learning. The general conclusion is that agile leadership may provide an enabling context for making agility a durable lever for the control of uncertainty, while CDM can be regarded as a structured and emerging form of agile management specifically oriented toward the progressive stabilization of software development under uncertainty. This study nevertheless presents several limitations. First, it is based on cross-sectional perceptual data collected from developers working in software development companies in Morocco, which limits both causal interpretation and the generalizability of the findings to other institutional or organizational contexts. Second, the analysis relies on self-reported Likert-scale measures and on a perceived improvement in uncertainty management rather than on objective project performance indicators such as delivery delays, defect rates, rework, or requirements volatility. Because predictor and outcome measures were collected from the same respondents at a single point in time, common method variance cannot be completely excluded. Future studies should combine multiple data sources, temporal separation, or objective project indicators to assess the robustness of the observed relationships. Third, the sample retained for the empirical

analysis remains limited, and the study does not compare CDM with other agile configurations or hybrid project management approaches. Finally, the empirical design does not capture the longitudinal dynamics through which uncertainty evolves across sprints and project phases. Future research could therefore extend the analysis through comparative, longitudinal, and multi-actor designs, while also examining more directly whether CDM, as a new form of agile management, improves observable project outcomes in different software development settings.

ACKNOWLEDGMENT

The authors would like to thank all software developers and industry professionals who voluntarily participated in the survey and contributed valuable insights to this study. Their participation made the empirical evaluation of the proposed Collaborative Development Method (CDM) possible.

REFERENCES

[1] Ahmed, J., Mrugalska, B., & Akkaya, B. (2022). Agile management and VUCA 2.0 (VUCA-RR) during Industry 4.0. In B. Akkaya, M. W. Guah, K. Jermsittiparsert, H. Bulińska-Stangrecka, & Y. Kaya (Eds.), Agile management and VUCA-RR: Opportunities and threats in Industry 4.0 towards Society 5.0 (pp. 13–26). Emerald Publishing Limited. https://doi.org/10.1108/978-1-80262-325-320220002

[2] Akkaya, B., & Ahmed, J. (2022). VUCA-RR toward Industry 5.0. In B. Akkaya, M. W. Guah, K. Jermsittiparsert, H. Bulińska-Stangrecka, & Y. K. Koçyiğit (Eds.), Agile management and VUCA-RR: Opportunities and threats in Industry 4.0 towards Society 5.0 (pp. 1–11). Emerald Publishing Limited. https://doi.org/10.1108/978-1-80262-325-320220001

[3] Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies (iJIM), 14(11), pp. 246–270. https://doi.org/10.3991/ijim.v14i11.13269

[4] Azonuche, T. I., & Enyejo, J. O. (2025). Adaptive risk management in agile projects using predictive analytics and real-time velocity data visualization dashboard. International Journal of Innovative Science and Research Technology, 10(4), 2032–2047. https://doi.org/10.38124/ijisrt/25apr2002

[5] Chen, J., Huang, R., Jiang, Y., Pu, C., Huang, X., Sun, X., & Liu, Y. (2023). ScrumOntoSPL: Collaborative method of agile product line engineering for software resource reuse. Electronics, 12(11), 2421. https://doi.org/10.3390/electronics12112421

[6] Choraś, M., Springer, T., Kozik, R., López, L., Martínez-Fernández, S., Ram, P., Rodríguez, P., & Franch, X. (2020). Measuring and improving agile processes in a small-size software development company. IEEE Access, 8, 78452–78466. https://doi.org/10.1109/ACCESS.2020.2990117

[7] Conforto, E. C., & Amaral, D. C. (2016). Agile project management and stage-gate model—A hybrid framework for technology-based companies. Journal of Engineering and Technology Management, 40, 1–14. https://doi.org/10.1016/j.jengtecman.2016.02.003

[8] Cooper, R. G. (2016). Agile–Stage-Gate hybrids. Research-Technology Management, 59(1), 21–29. https://doi.org/10.1080/08956308.2016.1117317

[9] Dey, S., & Lee, S.-W. (2021). Multilayered review of safety approaches for machine learning-based systems in the days of AI. Journal of Systems and Software, 176, Article 110941. https://doi.org/10.1016/j.jss.2021.110941

[10] Dobrowolski, Z., Ledzianowski, J., & Dobrowolska, M. (2021). Towards to agile management control systems at the university: Preliminary research. European Research Studies Journal, 24(1), 1220–1229. https://www.um.edu.mt/library/oar/handle/123456789/81395

[11] Dönmez, D., & Grote, G. (2018). Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities. Information and Software Technology, 93, 94–111. https://doi.org/10.1016/j.infsof.2017.08.015

[12] Ezghari, S., & Zahi, A. (2018). Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method. Applied Soft Computing, 67, 540–557. https://doi.org/10.1016/j.asoc.2018.03.022

[13] Fagerholm, F., & Pagels, M. (2014). Examining the structure of lean and agile values among software developers. In Agile Processes in Software Engineering and Extreme Programming (XP 2014) (pp. 218–233). Springer. https://doi.org/10.1007/978-3-319-06862-6_15

[14] Gheorghe, A.-M., Gheorghe, I. D., & Iatan, I. L. (2020). Agile software development. Informatica Economică, 24(2), 90–100. https://doi.org/10.24818/issn14531305/24.2.2020.08

[15] Gren, L., Torkar, R., & Feldt, R. (2017). Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. Journal of Systems and Software, 124, 104–119. https://doi.org/10.1016/j.jss.2016.11.024

[16] Gupta, R. K., Shukla, S., Rajan, A. T., & Aravind, S. (2022). Strategies for effective product roadmap development and execution in data analytics platforms. International Journal for Research Publication and Seminar, 13(1). https://jrps.shodhsagar.com/index.php/j/article/view/1515

[17] Hazzan, O., & Dubinsky, Y. (2014). Agile anywhere: Essays on agile projects and beyond. Springer Cham. https://doi.org/10.1007/978-3-319-10157-6

[18] Hendler, S. (2019). Digital-physical product development: A qualitative analysis. European Journal of Innovation Management, 22(2), 315–334. https://doi.org/10.1108/EJIM-01-2018-0026

[19] Highsmith, J. (2009). Agile project management: Creating innovative products (2nd ed.). Addison-Wesley. https://dl.acm.org/doi/10.5555/1594896

[20] Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. Computer, 34(9), 120–127. https://doi.org/10.1109/2.947100

[21] Khan, A. A., & Keung, J. (2016). Systematic review of success factors and barriers for software process improvement in global software development. IET Software, 10(5), 125–135. https://doi.org/10.1049/iet-sen.2015.0038

[22] Kläs, M., & Vollmer, A. M. (2018). Uncertainty in machine learning applications: A practice-driven classification of uncertainty. In B. Gallina, A. Skavhaug, E. Schoitsch, & F. Bitsch (Eds.), Computer safety, reliability, and security: SAFECOMP 2018 (pp. 431–438). Springer. https://doi.org/10.1007/978-3-319-99229-7_36

[23] Klünder, J., Karajic, D., Tell, P., Karras, O., Münkel, C., Münch, J., MacDonell, S. G., Hebig, R., & Kuhrmann, M. (2020). Determining context factors for hybrid development methods with trained models. In Proceedings of the International Conference on Software and System Processes (pp. 61–70). Association for Computing Machinery. https://doi.org/10.1145/3379177.3388898

[24] Kreye, M. E., Ramírez Hernández, T., & Eppinger, S. (2025). Translating agile management practices into a traditional industry context. Journal of Product Innovation Management, 42(2), 444–467. https://doi.org/10.1111/jpim.12760

[25] Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., Pfahl, D., et al. (2022). What makes agile software development agile? IEEE Transactions on Software Engineering, 48(9), 3523–3537. https://doi.org/10.1109/TSE.2021.3099532

[26] Levitt, R. E. (2011). Towards project management 2.0. Engineering Project Organization Journal, 1(3), 197–210. https://doi.org/10.1080/21573727.2011.609558

[27] Lill, P. A., Wald, A., & Gleich, R. (2020). Agility and the role of project: Internal control systems for innovation project performance. International Journal of Innovation Management, 24(7), Article 2050064. https://doi.org/10.1142/S1363919620500644

[28] Lishner, I., & Shtub, A. (2023). Enhancing strategic planning of projects: Selecting the right product development methodology. Information, 14(12), 632. https://doi.org/10.3390/info14120632

[29] Logue, K., & McDaid, K. (2008). Handling uncertainty in agile requirement prioritization and scheduling using statistical simulation. In

Agile 2008 Conference (pp. 73–82). IEEE. https://doi.org/10.1109/Agile.2008.79

[30] Makkass, M., Laghouaouta, Y., & Anwar, A. (2025). Agile method for collaborative development process by exploring use limits of Scrum. In M. Ben Ahmed, B. A. Abdelhakim, İ. R. Karaş, & K. Ben Ahmed (Eds.), Innovations in Smart Cities Applications Volume 8 (Lecture Notes in Networks and Systems, Vol. 1310, pp. 475–489). Springer. https://doi.org/10.1007/978-3-031-88653-9_47

[31] Makkass, M., Laghouaouta, Y., & Anwar, A. (2025). The role of agile leadership in team motivation within software development projects: A study case of multinational companies in Morocco. Journal of Computer Science, 21(11), 2735–2744. https://doi.org/10.3844/jcssp.2025.2735.2744

[32] Marinho, M., Noll, J., Richardson, I., & Beecham, S. (2019). Plan-driven approaches are alive and kicking in agile Global Software Development [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1906.08895

[33] Mendonça de Sá Araújo, C. M., Miranda Santos, I., Dias Canedo, E., & Favacho de Araújo, A. P. (2019). Design thinking versus design sprint: A comparative study. In A. Marcus & W. Wang (Eds.), Design, user experience, and usability. Design philosophy and theory (pp. 291–306). Springer. https://doi.org/10.1007/978-3-030-23570-3_22

[34] Nahar, N., Zhou, S., Lewis, G., & Kästner, C. (2022). Collaboration challenges in building ML-enabled systems: Communication, documentation, engineering, and process. In Proceedings of the 44th International Conference on Software Engineering (ICSE '22) (pp. 413–425). Association for Computing Machinery. https://doi.org/10.1145/3510003.3510209

[35] Perkusich, M., Chaves e Silva, L., Costa, A., Ramos, F., Saraiva, R., Freire, A., Dilorenzo, E., Dantas, E., Santos, D., Gorgônio, K., Almeida, H., & Perkusich, A. (2020). Intelligent software engineering in the context of agile software development: A systematic literature review. Information and Software Technology, 119, Article 106241. https://doi.org/10.1016/j.infsof.2019.106241

[36] Odzaly, E. E., Greer, D., & Stewart, D. (2018). Agile risk management using software agents. Journal of Ambient Intelligence and Humanized Computing, 9, 823–841. https://doi.org/10.1007/s12652-017-0488-2

[37] Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., Dar, S. H., & Khalid, S. (2021). Requirement engineering challenges in agile software development. Mathematical Problems in Engineering, 2021, Article 6696695, 1–18. https://doi.org/10.1155/2021/6696695

[38] Ries, E. (2011). How today's entrepreneurs use continuous innovation to create radically successful businesses. The lean startup. https://ohkliberec.cz/wp-content/uploads/2013/10/START-UP.pdf

[39] Salin, H., & Lundgren, M. (2022). Towards agile cybersecurity risk management for autonomous software engineering teams. Journal of Cybersecurity and Privacy, 2(2), 276–291. https://doi.org/10.3390/jcp2020015

[40] Schwaber, K., & Sutherland, J. (2020). Der visuelle Scrum Guide (November 2020). https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-German.pdf

[41] Sinha, A., & Das, P. (2021). Agile methodology vs. traditional waterfall SDLC: A case study on quality assurance process in software industry. In 2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech) (pp. 1–4). IEEE. https://doi.org/10.1109/IEMENTech53263.2021.9614779

[42] Tavares, B., da Silva, C. E. S., & de Souza, A. D. (2019). Risk management analysis in Scrum software projects. International Transactions in Operational Research, 26(5), 1884–1905. https://doi.org/10.1111/itor.12401

[43] Teece, D. J., Peteraf, M., & Leih, S. (2016). Dynamic capabilities and organizational agility: Risk, uncertainty, and strategy in the innovation economy. California Management Review, 58(4), 13–35. https://doi.org/10.1525/cmr.2016.58.4.13

[44] Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S. G., Münch, J., Pfahl, D., Linssen, O., & Kuhrmann, M. (2019). What are hybrid development methods made of? An evidence-based characterization. In Proceedings of the International Conference on Software and Systems Process (ICSSP 2019) (pp. 105–114). IEEE. https://doi.org/10.1109/ICSSP.2019.00022

[45] Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G., & Forcellini, F. A. (2018). Agile practices and the promotion of entrepreneurial skills in software development. Journal of Software: Evolution and Process, 30(6), e1945. https://doi.org/10.1002/smr.1945

[46] van Oorschot, K. E., Sengupta, K., & Van Wassenhove, L. N. (2018). Under pressure: The effects of iteration lengths on agile software development performance. Project Management Journal, 49(6). https://doi.org/10.1177/8756972818802714

[47] Waterman, M. (2018). Agility, risk, and uncertainty, Part 1: Designing an agile architecture. IEEE Software, 35(2), 99–101. https://doi.org/10.1109/MS.2018.1661335

[48] Włodarski, R., Poniszewska-Marańda, A., & Falleri, J. (2020). Comparative case study of plan-driven and agile approaches in student computing projects. In 2020 28th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE. https://doi.org/10.23919/SoftCOM50211.2020.9238196

[49] Zaslavska, K., & Zaslavskyi, V. (2024). Implementation of the project approach as a tool for ensuring flexibility and adaptability of business structures in a turbulent environment. Ukrainian Journal of Applied Economics and Technology, 2024(3), 181–185. https://doi.org/10.36887/2415-8453-2024-3-31

Smart IoT Robot Toy with AI-Based Safety Judgment and Parental Monitoring Features

Check-Yee Law[1], Wee-Yang Goh[2], Yong-Wee Sek[3]
COE for Advanced Cloud-Centre for Intelligent Cloud Computing (CICC), Multimedia University, Melaka, Malaysia[1, 2]
Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia[3]

***Abstract*—Smart Internet of Toys (IoToys) can provide interactive learning and companionship for children, but many systems still provide limited support for guardian awareness and safety notification. This study presents the design and development of a smart IoT robot toy with AI-based safety judgment and parental monitoring features. The prototype integrates a robot head unit, web-based monitoring dashboard, camera viewing, remote movement control, two-way communication, AI-assisted safety judgment, Telegram guardian notification, memory-based allergy reminder, chat history, and behavior report. The system uses an ESP32-S3 CAM head unit, an ESP32-C3 mobile base, a Node.js server, cloud-assisted AI services, and a Telegram Bot notification channel. Scenario-based testing was conducted, and the results showed that the prototype can support remote observation and notify guardians about child safety-related scenarios such as fear, injury, danger, and food allergies.**

Keywords*—Child safety; internet of toys; IoToy; IoT robot; parental monitoring; privacy concerns; smart toy; Telegram notification

I. Introduction

Smart toys have progressed from being simple electronic devices to complex connected systems that are able to sense, communicate, and act upon users. These developments play a key role in the domain of the Internet of Toys (IoToys), which aims to have toys interact with children and their environments, typically autonomously. These toys are equipped with sensors, actuators, and AI algorithms that allow them to offer tailored and interactive play experiences [1, 2].

Smart toys can be used as a kind of entertainment and learning instrument in home scenarios, which can create creative play and cognitive development activities [3, 4]. They can gather and analyze data using cameras, microphones, and other sensors. However, there is a problem of privacy and safety concerns with children interacting with these toys without the supervision of their parents [1, 2, 5]. Some toys, for example, stream data to servers on an ongoing basis, which could allow audio surveillance, whereas others have been the victim of privacy breaches or the ability of the attacker to control the toy [2, 5].

Therefore, it is necessary to improve smart toys to overcome the above challenges, which should include functions to entertain and help the guardians keep track of safety-related events. This includes receiving live alerts for any strange activity or possible dangers identified by the toy's sensors or AI-based analysis.

However, existing research reveals a gap in a unified security model or security guidelines for secure use of smart toys [2, 6]. The work on ontologies that explicitly support privacy and security requirements for smart toys is still in progress, and implementation is still limited [1, 7]. This implies that smart toys have great potential to be interactive and educational, but they need to be designed with careful consideration of strong safety and privacy protections to support guardians' ability to keep children safe in their interactions with the toys when they are not being supervised [1, 2, 5].

There are some smart toys that focus on entertainment, learning, or interaction, with limited monitoring by guardians and safety notifications. For example, some devices feature mobile apps, but they tend to focus more on content control instead of situation awareness and parental control [1]. Previous studies have also pointed out that child-centered IoT toys need to be more attentive to privacy, security, and parental control, as well as communication safety. Smart toys are often equipped with sensors, cameras, and microphones that gather personal and situational information, a fact that is worrying many about their ability to be accessed by other people without consent, and their potential to violate privacy rights [1, 2]. In addition, the absence of uniform guidelines for parental control and privacy protection adds to these risks, making children susceptible to digital and physical dangers [8, 9].

This study reports on the parental monitoring part of an AI-enabled IoT robot toy developed as a prototype. The aim is not to replace adult supervision, but to provide an additional awareness and communication channel between the child, the robot, and the guardian. Compared with the complete robot platform, this study focuses on modules related to monitoring, safety notification, remote intervention, and behavior awareness.

The remainder of this study is organized as follows. Section II reviews related work on Internet of Toys, smart toy safety, and parental monitoring. Section III describes the proposed system design and the parental monitoring workflow. Section IV describes the implementation of the web-based monitoring dashboard, camera monitoring, Telegram notification, AI-based judgment, behavior report, and remote intervention modules. Section V presents the scenario-based testing results with screenshots and photos as supporting evidence. Section VI discusses the findings, limitations, and privacy concerns of the prototype. Finally, Section VII concludes the research work with the main contributions and future work.

II. Related Work

Internet-connected toys (IoToys) combine embedded hardware, sensors, wireless communications, and software services to provide interactive and connected play experiences. The toys are typically designed with integrated capabilities like voice control, wireless connectivity, and adaptive feedback, which add to their attractiveness and usability at home [10, 11, 12]. Even at home, many factors come into play, including safety, transparency, and having guardians' control. The main issues related to safety involve privacy and security related to the collection and processing of personal data, such as sensitive data about children, by IoToys. This information may be subject to hacking, unauthorized use, or misuse, raising serious privacy concerns [10, 13, 14, 15]. Another major concern is transparency; information regarding the collection, storage, and use of personal data by these toys is often unclear to children and their guardians. To enhance transparency, privacy rules and privacy education resources are being developed to better inform children and parents about data collection mechanisms [11, 16].

Guardian control is vital to making sure that IoToys are used safely and responsibly. There are currently challenges related to the lack of uniform safety labelling on IoToys and the difficulty for parents to be able to determine the level of security that these devices have before buying them. This is the reason why there are privacy labels and security labels such as "ChildShield" being proposed to help consumers understand the safety features of IoToys [14]. Furthermore, IoToys will need to be both educational and entertaining, yet safety measures must be very robust to ensure that they are interesting and safe for children [10, 12]. This means that IoToys creates innovative and personalized playing experiences, but its use at home will require attention to privacy, security, and parental control issues to ensure safe and transparent use [10, 11, 13, 14, 15].

Smart toys pose serious security and privacy concerns related to children's speaking data and cloud processing, network communication, application vulnerabilities, and lack of parental control. Smart toys can also record voice information and send it to the cloud to analyze. This adds a layer of privacy concerns, since live streams of data can be exploited for unauthorized audio monitoring. For example, some AI-powered toys are known to send constant audio data to servers, which might pose privacy and surveillance concerns [5, 17]. There are many smart toys that have security issues with their network communications with servers, so they can be hacked and their data exposed. These weaknesses tend to be related to bad practices in the implementation of security policies, as has been reported in the analyses of toy communication environments [18, 1, 19]. Smart toy companion apps often have excessive permissions and do not employ robust security measures. Static and dynamic analyses have identified problems in these apps, including unnecessary data collection and lack of adherence to privacy laws [19, 2]. Parents' control mechanisms are still inadequate. This adds to privacy concerns because parents may not have the means to monitor or limit their children's data collection and utilization by toys. This is reported to be a major concern of parents, especially in developed economies [20]. Besides that, with their capacity to collect and handle delicate data like location and activity patterns, smart toys can encounter information leaks, unauthorized access, and even physical safety hazards. In addition to this, privacy frameworks are not standardized, and compliance is inconsistent with privacy laws [21, 22, 23]. This indicates that the security and privacy issues with smart toys are complex and encompass technical, regulatory, and usability concerns. Solving these problems needs better security measures, stricter enforcement of regulations, and better parental control options.

McReynolds et al. [24] reported that Internet-connected toys may record or transmit children's speech, while parents and children may not always understand what data are collected or how the data are used. Chu et al. [18] found that commercially available children's IoT toys can contain network and application vulnerabilities that conflict with privacy and security best practices. Shasha et al. [2] further proposed a threat taxonomy for smart toys and showed that unsafe access or weak protection may expose children to privacy leakage and safety risks.

Parental control for smart toys should be accompanied by visibility of children's interactions, control over data sharing, notification of important events, and means for making children's interactions known to their guardians. Parents should have resources to watch and comprehend their children's interactions with their smart toys. This means access to the information gathered and its usage [25, 8, 9]. For instance, dashboards or interfaces to show parents what their child is doing and sharing with the toy interaction information are crucial [26].

Parental control tools need to enable parents to control and limit information that their child shares with their smart toy or third parties. This is important because of potential privacy violations and misuse of data [25, 8, 27, 21]. Tools available today are inadequate to meet these needs, and having customizable data-sharing preferences [8, 28] is therefore important. If the parents are involved in the sharing of sensitive information, or if there is a possibility that their information is being shared in an insecure way, they should be informed of this. This will enable prompt action to help protect the child's privacy and safety [1, 28].

Tools should help increase parental awareness of privacy risks and offer easy-to-use interfaces for effective privacy management by non-technical users [29, 30]. Parents must have control of what happens to their children, which means making the design of the products more usable and having clear privacy labels. This suggests that there is strong support for the proposed parental control measures, which include visibility, control, notification, and awareness mechanisms to address concerns about the privacy and safety of smart toys. Therefore, monitoring and notification should be considered as practical design requirements for child-oriented IoT systems.

Based on these concerns, the proposed system integrates parental monitoring directly into the robot platform. Instead of only providing a separate mobile application, the system connects the robot, server, web dashboard, AI judgment, and Telegram Bot notification into one workflow. This allows the guardian to easily receive important information without continuously checking the dashboard.

III. System Design

A. Overall Architecture

Fig. 1 shows the context diagram of the parental monitoring workflow of the proposed IoT toy. The parental monitoring architecture consists of a robot head unit, a server layer, an AI judgment component, a web dashboard, a behavior logging module, and a Telegram notification module. The robot collects interaction input through microphone and camera modules, while the server coordinates AI processing, dashboard communication, logging, and notification. The guardian accesses the system through the web dashboard or receives Telegram alerts when serious situations, such as injury reports, fear expressions, danger-related messages, allergy-related requests, or direct requests to contact the guardian, are detected.

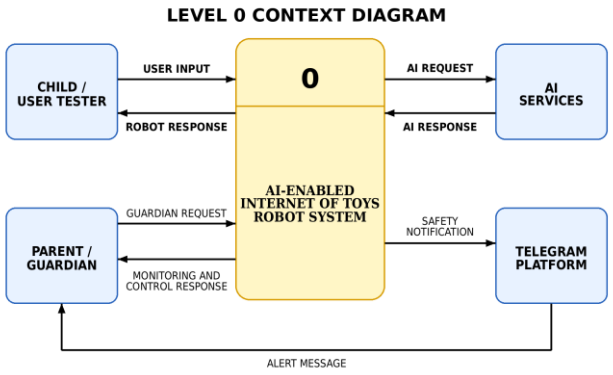


Fig. 1. Context diagram of the parental monitoring workflow.

B. Hardware and Server Components

The robot head unit is based on an ESP32-S3 CAM module and includes a camera, microphone, speaker output, touch input, and display feedback. The mobile base uses an ESP32-C3 controller for wheeled movement. The two parts form a modular robot platform. The Node.js server acts as the central coordination layer between the robot, browser interface, AI services, memory storage, report module, and Telegram Bot.

C. Parental Monitoring Workflow

During normal interaction, the child can communicate with the robot using natural language speech instead of a fixed command list. The interaction can be started through the robot's wake-up mechanism, such as voice activation or touch activation. After activation, the microphone captures the child's speech and sends the audio to the server for speech-to-text processing. The recognized text is then forwarded to the GPT API to generate a context-aware response. Since the GPT API supports multilingual text processing, the robot can support multilingual interaction in principle. However, the actual language performance depends on the speech-to-text module, the selected GPT model, the text-to-speech output, and the availability of support for low-resource languages.

Visual assistance refers to the use of the ESP32-S3 camera to support image-based interaction. For example, when the child asks a visual question such as "What is this?" or asks the robot to look at an object, the camera captures the current image and sends it together with the child's question to the AI service. The AI analyses the visual context and generates a response, which is then converted into speech and played through the robot speaker.

The AI judgment module is executed after each child-robot interaction. The server provides the module with the child's original input, the AI-generated response, and the optional camera image or visual description. The module then classifies the interaction into two outcomes: "guardian notification required" or "no guardian notification required". A notification is triggered when the interaction contains a direct request to contact a guardian, emotional distress, injury, sickness, danger, stranger-related concern, fire or electrical risk, unsafe food, allergy-related request, or other safety-related condition. If the interaction is a normal conversation, such as asking for a story or asking a general question, the system stores the interaction in the behavior log without sending a Telegram alert.

When a notification is required, the server generates a short summary of the situation and sends it to the guardian through Telegram. The Telegram message includes the child's original input, the AI response, the situation summary, whether an image is attached, and the timestamp. This decision mechanism reduces unnecessary alerts while still notifying the guardian during meaningful safety-related situations.

IV. Implementation

A. Web-Based Monitoring Dashboard

The web-based monitoring dashboard, as shown in Fig. 2, allows the guardian to access the robot through a browser and interact with different monitoring functions. For camera viewing, the guardian clicks the camera or stream button to view the child's surrounding environment through the ESP32-S3 camera. The output is a live or updated camera view, which helps the guardian understand the child's current environment when the guardian is not physically beside the child.

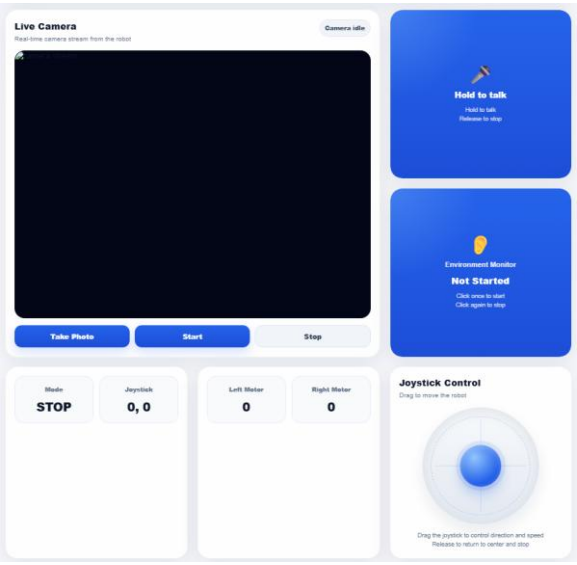


Fig. 2. Web dashboard for camera viewing and remote robot control.

For movement control, the guardian presses the forward, backward, left, or right buttons on the interface. The command is sent from the browser to the server and then forwarded to the ESP32-C3 mobile base. The output is the physical movement of

the robot, allowing the guardian to reposition the robot or observe a different direction remotely.

For AI wake-word control and camera toggle, the guardian can enable or disable selected robot functions from the dashboard. The output is the updated function status shown on the interface. This helps the guardian control when the robot should listen for interaction or activate the camera, depending on the usage situation.

The dashboard also provides links to memory, chat history, and behavior report pages. Through these pages, the guardian can review previous child-robot conversations, safety-related events, and reminders stored, such as allergy information. These outputs help the guardian understand what the child asked, how the robot responded, and whether any important behavior or safety condition occurred.

B. Telegram Guardian Notification

Telegram Bot is used as the guardian notification channel because it supports fast message delivery and can include both text and images. When the system identifies a safety-related event or a guardian contact request, the server sends a structured message to the guardian. The message can include the child's original words, the AI response, a brief situation summary, image availability, and time information. Fig. 3 illustrates an example of the Telegram message generated for a child-to-guardian communication request.

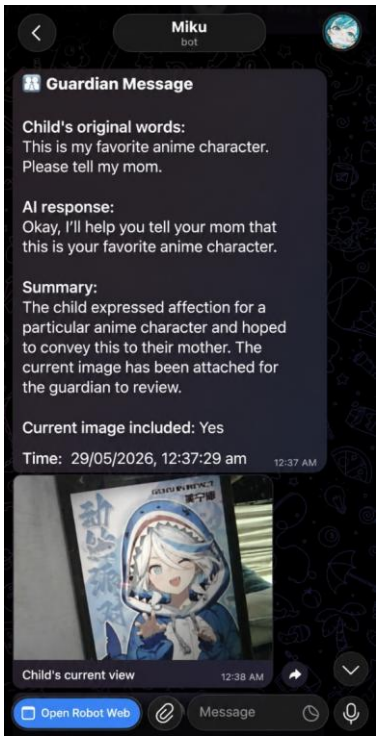


Fig. 3. Telegram notification generated for a child-to-guardian communication request.

C. AI Safety Judgment and Behavior Logging

The AI judgment component operates after each child-robot interaction. The server sends the child's original input, the AI-generated response, and optional visual context from the camera to a judgment prompt. The judgment prompt asks the AI to classify the interaction into two outcomes: guardian notification required or no guardian notification required. The decision is based on whether the interaction contains safety-related or guardian-related content, such as direct requests to contact a parent, expressions of fear such as "I am scared," injury reports such as "I fell down," dangerous environment descriptions such as "There is fire," and allergy-related questions such as "Can I eat peanuts?"

If guardian notification is required, the system generates a short summary and sends a structured Telegram message to the guardian. The message includes the child's original input, the AI response, the event summary, image availability, and timestamp. If the interaction is a normal conversation, such as asking for a story or asking a general question, the system stores the record in the behavior log without sending a Telegram alert. This mechanism helps reduce unnecessary notifications while still allowing the guardian to receive important safety-related or behavior-related information.

The system also records selected chat history and behavior information so that the guardian can review important events later. Fig. 4 shows the behavior report page used to review previous child-robot interactions and safety-related records. Through this report, the guardian can understand what the child said, how the robot responded, and whether any event required attention.

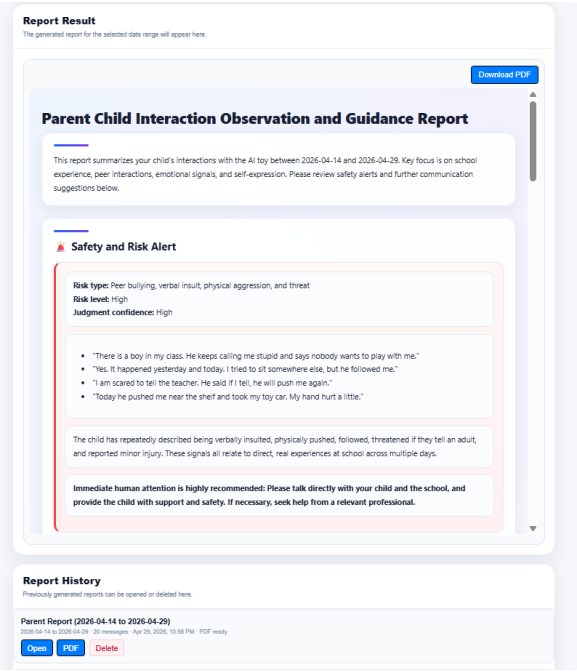


Fig. 4. Behavior report page for reviewing interaction and safety-related events.

D. Allergy Reminder and Remote Intervention

The memory module can store simple child-related information, such as allergy reminders. When the child asks about food that may be unsafe, the system retrieves the stored allergy information and includes it in the AI response. The robot then responds with a warning through speech output, reminding the child to avoid the restricted food. An example of the allergy reminder output is shown in Fig. 5.

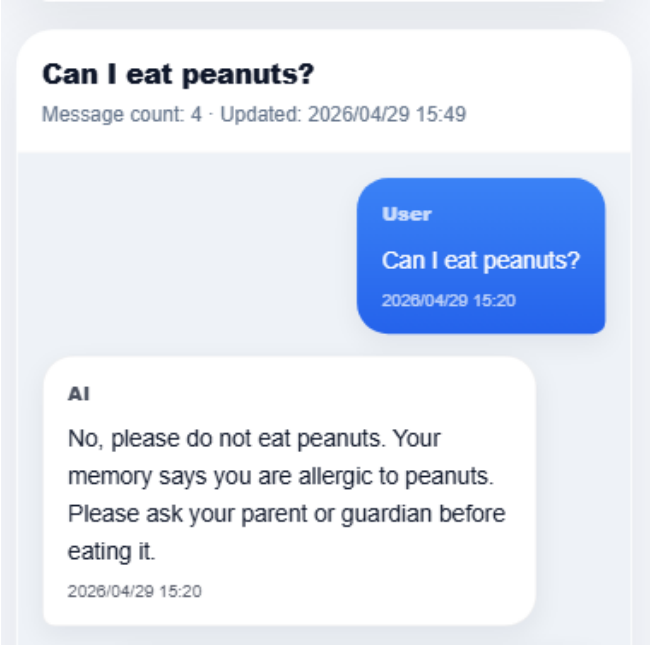


Fig. 5. Allergy reminder output generated based on stored child-related information.

Remote intervention is supported through camera viewing, movement control, and two-way guardian communication. The guardian can first check the camera view to understand the child's surrounding environment. If the guardian needs a different viewing angle, the guardian can use the movement buttons on the dashboard to reposition the robot. If the child needs guidance, the guardian can use the hold-to-talk function to speak through the web interface, and the robot plays the voice message through its speaker to guide the child remotely. Fig. 6 shows the hold-to-talk function used for remote intervention.

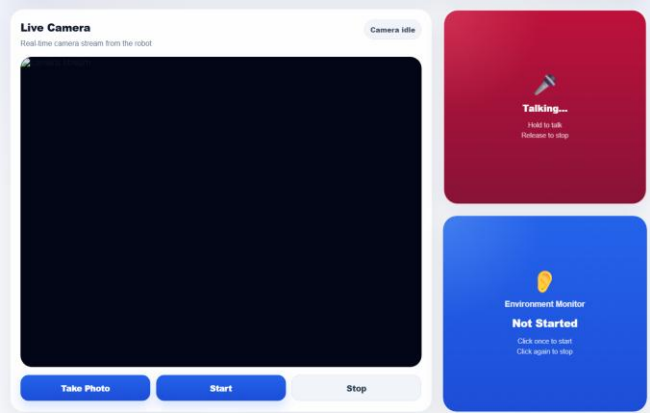


Fig. 6. Two-way guardian communication interface for remote intervention.

V. Scenario-Based Testing and Results

Scenario-based testing was conducted to verify whether the parental monitoring functions produced the expected outputs. The testing focused only on monitoring-related features. Each test used a realistic scenario, input, and output, as shown in Table I. The scenarios include guardian contact request, fear expression, injury report, danger warning, allergy reminder, remote monitoring, remote intervention, and behavior review.

TABLE I. Scenario-Based Testing of Parental Monitoring Features

Scenario	Input	Output
Guardian contact	Child asks robot to tell mother	Telegram message is sent to guardian
Fear expression	Child says "I am scared"	AI comforts child and alerts guardian
Injury report	Child says "I fell down"	Safety notification is generated
Danger warning	Child says "There is fire"	Warning response and Telegram alert generated
Allergy reminder	Child asks about restricted food	Robot gives allergy warning
Remote monitoring	Guardian opens dashboard	Camera view and control buttons are displayed
Remote intervention	Guardian uses hold-to-talk function	Robot delivers guardian voice guidance to the child
Behavior review	Guardian opens report page	Interaction summary is shown

The supporting screenshot evidence is shown in Fig. 2 for the web-based monitoring dashboard, Fig. 3 for Telegram guardian notification, and Fig. 4 for behavior report review. Additional output evidence is provided in Fig. 5 for the allergy reminder and Fig. 6 for two-way guardian communication. These figures provide supporting evidence for the main outputs as listed in Table I.

The test results show that Telegram notifications were triggered for guardian-contact and safety-related scenarios, while the web dashboard supported remote camera viewing, movement control, and remote intervention. The allergy reminder and behavior report pages also helped the guardian review child-related information and previous interaction records. These results indicate that the proposed system can support guardian awareness and remote intervention in basic child-care scenarios.

VI. Discussion

The main advantage of the system is that the guardian does not need to constantly monitor the web dashboard. Safety-related information can be pushed through Telegram when the AI judgment module detects a meaningful event. Camera viewing, behavior reporting, and remote control provide additional context for the guardian to understand the child's situation and respond when necessary. These features allow the IoT toy to act not only as an interactive robot, but also as an additional awareness and communication channel between the child and the guardian.

The reliance on Connectivity and AI Judgment is a recurring theme, with abstracts underscoring the vulnerabilities introduced by connections with other systems, including cloud-based ones, and on real-time decision-making capabilities [31, 32]. This is in line with the issue of relying on cloud AI and Wi-Fi. There are multiple studies that highlight some of the problems with the accuracy of AI, such as false negatives and false positives. In medical and child well-being, the use of AI systems can be complicated by problems with predictive

accuracy, especially if qualitative or contextual data is lacking [32, 33]. This can result in false alarms or in failure to detect the alarm, as mentioned in the question. AI systems need to be thoroughly tested and have more precise instructions on how to operate to become more reliable. For instance, the need for robust evaluation metrics, post-deployment monitoring, and dynamic testing scenarios is frequently mentioned [34, 35]. These measures can be helpful to improve the safety classification rules and minimize errors.

The privacy issues surrounding the IoToy, including the capturing of sensitive information such as children's voices, images captured by the camera, notes of interactions, and any alerts given, were raised. Currently, it is only a prototype where access to this data is limited to a configured guardian through a web dashboard and a Telegram account, but other security elements such as more robust authentication, encrypted communication, access control, and data retention settings are needed before deployment with children.

IoT devices, including smart toys, often collect sensitive data such as video, audio, and interaction logs, which can expose children to significant privacy risks if not adequately secured [36, 37, 38]. For example, there have been reports of weaknesses in IoT-enabled toys that have led to the unauthorized use of cameras and sensors to capture live images and videos without the user's consent [38, 39, 40]. The current IoToy prototype using the web dashboard and Telegram account to access data emphasizes the need for a more secure approach. Research highlights the need for the use of encryption, multi-factor authentication, and access control to reduce risks of unauthorized access and data breaches [36, 39, 41]. For instance, blockchain authentication and encryption have been suggested to improve the security of IoT toys. One concern is Data Retention and Regulatory Gaps, where IoT devices may not have explicit data retention policies, and there is a lack of transparency regarding who accesses or stores sensitive data. To conclude, although the IoToy prototype restricts data access to guardians, there is a strong need for further improvement in authentication, encryption, and access control to ensure effective protection of children's privacy. Moreover, it is essential to address the regulatory gaps and put in place strong data retention policies to guarantee long-term privacy and security [36, 37, 39, 41].

VII. Conclusion

This study presented the design and development of a smart IoT toy with parental monitoring features. The main contribution of this work is the development of a server-assisted IoT toy prototype that connects child-robot interaction with parental monitoring functions. The proposed IoT toy not only provides a joyful play experience and entertainment, but it also helps guardians receive important safety-related information about the child's behavior and surrounding environment. The prototype integrates web-based camera monitoring, remote movement control, AI-based safety judgment, Telegram guardian notification, allergy reminder, behavior logging, and behavior review to support parental monitoring and childcare. Scenario-based testing showed that the system can generate appropriate monitoring outputs, notify guardians, and support remote observation in safety-related situations. Future work will focus on improving notification accuracy, privacy protection, network robustness, and larger-scale user evaluation with parents and child users.

References

[1] L. G. de Carvalho and M. M. Eler, "Security requirements and tests for smart toys," Lecture Notes in Business Information Processing, pp. 291 - 312, 2018.

[2] S. Shasha, M. Mahmoud, M. Mannan and A. Youssef, "Playing with danger: A taxonomy and evaluation of threats to smart toys," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2986 - 3002, 2019.

[3] P. Ihamäki and K. Heljakka, "The internet of toys, connectedness and character-based play in early education," Advances in Intelligent Systems and Computing, pp. 1079 - 1096, 2019.

[4] T. Jayaraman , P. Rajkumar, M. Shakunthala , M. Vijayakumar , M. Vimal Kumar and N. Hemavijayrajar, "Eco-friendly IoT-based smart toy automation for interactive learning and engagement," Proceedings of the 6th International Conference on Electronics and Sustainable Communication Systems, pp. 1511 - 1515, 2025.

[5] V. Pavliv , N. Akbari and I. Wagner , "AI-powered smart toys: interactive friends or surveillance devices?," 2025.

[6] L. Pontes, G. Coutinho, P. C. K. Hung and B. Yankson, "Security in smart toys: A systematic review of literature," 2019.

[7] O. de Paula Albuquerque, P. C. K. Hung, S. M. Peres, F. Iqbal and M. Fantinato, "A preliminary design of privacy ontology for smart toys," in Proceedings of the Annual Hawaii International Conference on System Sciences, 2024.

[8] P. C. K. Hung, M. Fantinato and L. Rafferty, "A study of privacy requirements for smart toys," in Pacific Asia Conference on Information Systems, PACIS 2016 - Proceedings, 2016.

[9] P. C. K. Hung, F. Iqbal, S.-C. Huang, M. Melaisi and K. Pang, "A glance of child's play privacy in smart toys," in Lecture Notes in Computer Science, 2016, pp. 217-231.

[10] K. Nikolopoulou, "Internet of toys for young children: Educational value or threat?," in Research Anthology on Early Childhood Development and School Transition in the Digital Era, 2022, pp. 85-100.

[11] L. Rafferty, P. C. K. Hung, M. Fantinato, S. M. Peres, F. Iqbal, S.-Y. Kuo and S.-C. Huang, "Towards a privacy rule conceptual model for smart toys," in Proceedings of the Annual Hawaii International Conference on System Sciences, 2017.

[12] A. Catala, C. Sylla, A. G. Ozgur, P. Ihamäki and K. Heljakka, "Smart toys, smart tangibles, robots and other smart things for children," in Extended Abstracts - Proceedings of the 2020 ACM Interaction Design and Children Conference, IDC 2020, 2020.

[13] E. Amengual, A. Bibiloni, M. Mascaró and P. Palmer-Rodríguez, "What does the GDPR mean for IoToys data security and privacy?," in Advances in Intelligent Systems and Computing, 2019, pp. 881-889.

[14] S. Allana and S. Chawla, "ChildShield: A rating system for assessing privacy and security of internet of toys," Telematics and Informatics, 2021.

[15] S. Chaudron, D. Geneiatakis, I. Kounelis and R. Di Gioia, "Testing internet of toys designs to improve privacy and security," in Studies in Childhood and Youth, 2019, pp. 223-239.

[16] A. Manches and L. Plowman, "Smart toys and children's understanding of personal data," International Journal of Child-Computer Interaction, 2021.

[17] V. Pavliv, L. Lazri, J. Büchele and I. Wagner, ""Why is the sky blue?": On the feasibility of privacy-friendly conversational LLM smart toys," in Lecture Notes in Computer Science, 2026, pp. 49-65.

[18] G. Chu, N. Apthorpe and N. Feamster, "Security and privacy analyses of internet of things children's toys," IEEE Internet of Things Journal, pp. 978-985, 2019.

[19] M. Mahmoud, M. Z. Hossen, H. Barakat, M. Mannan and A. Youssef, "Towards a comprehensive analytical framework for smart toy privacy practices," in ACM International Conference Proceeding Series, 2018.

[20] F. M. Pinheiro Amâncio, A. P. Souza, M. Fantinato, S. M. Peres, P. C. K. Hung, L. G. Coutinho do Rêgo and J. Roa, "Parental perception of

children's privacy in smart toys in countries of different economic levels," Technology in Society, 2023.

[21] L. G. de Carvalho and M. M. Eler, "Security tests for smart toys," in ICEIS 2018 - Proceedings of the 20th International Conference on Enterprise Information Systems, 2018.

[22] O. D. P. Albuquerque, M. Fantinato, J. Kelner and A. P. de Albuquerque, "Privacy in smart toys: Risks and proposed solutions," Electronic Commerce Research and Applications, vol. 39, 2020.

[23] P.-C. Lin, B. Yankson, Z. Lu and P. C. K. Hung, "Children privacy identification system in LINE chatbot for smart toys," in IEEE International Conference on Cloud Computing, CLOUD, 2019.

[24] E. McReynolds, S. Hubbard, T. Lau, A. Saraf, M. Cakmak and F. Roesner, "Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems," 2017.

[25] O. D. P. Albuquerque, M. Fantinato, M. M. Eler, S. M. Peres and P. C. K. Hung, "A study of parental control requirements for smart toys," in Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 2020.

[26] M. L. Jones and K. Meurer, "Can (and should) Hello Barbie keep a secret?," in 2016 IEEE International Symposium on Ethics in Engineering, Science and Technology, ETHICS 2016, 2016.

[27] L. G. de Carvalho, M. Fantinato and M. M. Eler, "Security requirements identification and prioritization for smart toys," Electronic Commerce Research and Applications, 2020.

[28] O. de Paula Albuquerque, M. Fantinato, P. C. K. Hung, S. M. Peres, F. Iqbal, U. Rehman and M. U. Shah, "Recommendations for a smart toy parental control tool," Journal of Supercomputing, pp. 11156-11194, 2022.

[29] A. de Lima Salgado, R. P. de Mattos Fortes, R. R. de Oliveira and A. P. Freire, "Usability heuristics on parental privacy controls for smart toys: From an exploratory map to a confirmatory research," Electronic Commerce Research and Applications, 2020.

[30] A. de Lima Salgado, F. S. Dias, J. P. R. Mattos, R. P. de Mattos Fortes and P. C. K. Hung, "Smart toys and children's privacy: Usable privacy policy insights from a card sorting experiment," in SIGDOC 2019 - Proceedings of the 37th ACM International Conference on the Design of Communication, 2019.

[31] S. Vignesh, D. Sabarevason and A. R. Sachin, "IoT-enabled smart eyewear: A real-time platform for comprehensive child monitoring and proactive intervention," in 3rd International Conference on Emerging Applications of Material Science and Technology, ICEAMST 2025, 2025.

[32] P. van Dommelen, Y. C. S. Frans, E. Vlasblom, J. Bouwman and J. A. de Wilde, "AI in childhood development monitoring: Attitudes, needs, and concerns in Dutch youth healthcare," in Studies in Health Technology and Informatics, 2025, pp. 158-162.

[33] S. Kim and Y. Yoon, "Child protection in the digital age in South Korea: Professionals' perceptions on artificial intelligence (AI) and big data efficacy and ethical challenges," Child and Family Social Work, 2026.

[34] G. Ragone, S. Tibidò, P. Buono, R. Lanzilotti, A. Ragone and M. T. Baldassarre, "Evaluating the gaps in existing AI ethics frameworks for children," in CHItaly 2025 - Proceedings of the 16th Biannual Conference of the Italian SIGCHI Chapter, 2025.

[35] Y. Wang, W. Fu, Y. Zhang, D. Wang, Y. Gu, W. Wang, H. Xu, X. Ge, C. Ye, J. Fang, L. Su, J. Wang, W. He, X. Zhang and R. Feng, "Constructing and implementing a performance evaluation indicator set for artificial intelligence decision support systems in pediatric outpatient clinics: An observational study," Scientific Reports, 2024.

[36] D. Rivera, A. Garcia, M. L. Martin-Ruiz, B. Alarcos, J. R. Velasco and A. Gomez Oliva, "Secure communications and protected data for a internet of things smart toy platform," IEEE Internet of Things Journal, pp. 3785-3795, 2019.

[37] E. Haber, "The internet of children: Protecting children's privacy in a hyper-connected world," University of Illinois Law Review, pp. 1209-1248, 2020.

[38] J. Streiff, O. Kenny, S. Das, A. Leeth and L. J. Camp, "Poster abstract: Who's watching your child? Exploring home security risks with smart toy bears," in Proceedings - ACM/IEEE International Conference on Internet of Things Design and Implementation, IoTDI 2018, 2018.

[39] J. Streiff, S. Das and J. Cannon, "Overpowered and underprotected toys empowering parents with tools to protect their children," in Proceedings - 2019 IEEE 5th International Conference on Collaboration and Internet Computing, CIC 2019, 2019.

[40] J. Streiff, N. Noah and S. Das, "A call for a new privacy & security regime for IoT smart toys," in 5th IEEE Conference on Dependable and Secure Computing, DSC 2022 and SECSOC 2022 Workshop, PASS4IoT 2022 Workshop SICSA International Paper/Poster Competition in Cybersecurity, 2022.

[41] A. Alkhatib, L. Albdour, S. Fayyad and H. Ali, "Blockchain-enhanced multi-factor authentication for securing IoT children's toys," International Journal of Computational and Experimental Science and Engineering, pp. 1041-1049, 2024.

Real-Time Anomaly Detection Using ML, DL, and XAI for Financial Fraud: A Systematic Review

Padilla-Lopez Jimmy Andres, Lazo-Ordoñez Pedro Antonio, Juan Jesús Soria-Quijaite
Escuela de Ingeniería de Sistemas, Universidad Tecnológica del Perú, Lima, Perú

***Abstract*—Digital payments, fintech, and online banking are now part of our daily lives. Making a transfer, paying for a service, or shopping online is faster, but this progress has also given rise to new forms of fraud. Rule-based systems remain useful when the pattern is already known; the problem arises when attackers change their tactics. For this reason, this study reviews the use of machine learning, deep learning, and Explainable Artificial Intelligence to detect anomalies and prevent fraud in digital payment systems. The research was conducted as a systematic literature review. PICOC helped define the scope and search terms, while PRISMA guided the selection of articles. Scopus, Web of Science, and ScienceDirect were consulted. After applying the inclusion, exclusion, and methodological evaluation criteria, 33 studies were retained. The articles addressed cases of card fraud, digital banking, blockchain, money laundering, and other suspicious transactions. The most frequently mentioned models were Random Forest, XGBoost, SVM, Isolation Forest, LSTM, CNN, autoencoders, and graph networks. Hybrid approaches and tools such as SHAP and LIME also appeared, used to show which variables influenced an alert. Several studies reported good results in terms of accuracy, recall, and F1 score. However, challenges remain regarding synthetic data, class imbalance, limited information on latency, and insufficient validation with financial specialists. Their performance still needs to be tested with real-world data and under conditions similar to those of a financial institution.**

Keywords*—Financial fraud; machine learning; deep learning; XAI; digital payments

I. Introduction

Electronic payments, fintech, and online banking are part of everyday financial services. Their growth has sped up many transactions, but it has also opened up new opportunities for fraud. This affects businesses through financial losses and also affects users, who may lose trust in these platforms.

Aparcana-Tasayco et al.[1] note that the rise of connected digital services necessitates improvements in anomaly detection methods.

The reason is that forms of fraud do not remain static: they evolve, test new strategies, and seek to circumvent existing controls. Consequently, methods based on static rules prove insufficient for detecting complex fraudulent behavior. In light of this situation, anomaly detection has emerged as a key approach, enabling the identification of unusual patterns in large volumes of data. In this context, Zhang et al.[2] note that machine learning models based on probabilistic approaches improve the early detection of suspicious activities in financial systems.

Likewise, the use of machine learning techniques has enabled the development of predictive models capable of classifying financial transactions by learning from historical patterns. These models include supervised and unsupervised algorithms, which have proven to be more efficient than traditional methods in identifying fraud. In this regard, Kumar et al. [3] note that the use of deep learning techniques has significantly increased detection capabilities by enabling the modeling of complex and nonlinear relationships in financial data.

The digitization of payment systems has led to more sophisticated forms of financial fraud, which have outpaced traditional rule-based methods. In this context, Wang et al. (2024) propose the use of dual-channel memory networks to extract features from credit review texts, demonstrating that it is possible to improve the accuracy of fraud risk detection by leveraging unstructured data [4].

Digital payment systems require accurate models capable of processing large volumes of transactions in real time. Raja et al. (2025) propose a hybrid model that combines GraphSAGE with LSTM networks to detect suspicious patterns, achieving 95.4% accuracy in money laundering detection. [5]

Finally, hybrid models that combine machine learning and deep learning techniques have proven to be a promising alternative for improving accuracy and reducing false positives in fraud detection. [6]

A. Problem Definition

1) Current situation: Digital payments and fintech platforms have become a common part of financial activities due to the speed and convenience they offer. However, the increase in electronic transactions has also created more opportunities for fraud. In the United States alone, more than 2.7 billion electronic transactions are recorded each year, while organizations lose nearly 5% of their annual revenue due to this problem [7].

Detecting these crimes is not easy. Attackers combine cyber techniques with social engineering and constantly change their methods. At the same time, a system that is too sensitive may block legitimate transactions. Added to this is the need to protect user privacy and explain how automated decisions are made [8].

Traditional systems operate using manually defined rules and thresholds. The problem is that these rules must be updated every time a new form of fraud emerges, so they often react too

late. Černevičienė and Kabašinskas [9] explain that this reliance on fixed rules reduces adaptability. Furthermore, these systems can generate between 85% and 99% false positives, which affects customers and increases the workload for the teams responsible for reviewing alerts [10].

Another limitation lies in response time. Some systems still process transactions in batches, meaning that fraud may be detected after the transaction has already been completed [11]. This delay, coupled with the rise in attacks, continues to erode user confidence in digital financial services [12].

Deep learning models offer better capabilities for recognizing complex patterns, but they also present challenges. Their real-time application requires significant processing power, especially when they must analyze thousands of transactions in just a few seconds. Furthermore, fraud accounts for a very small portion of total transactions—in some cases, less than 0.1%. For this reason, a model that performs well in laboratory tests does not always maintain the same performance when applied on a large scale [13].

B. Rationale

Fraud in digital payments has a direct impact on businesses and the people who use these services. It is estimated that organizations lose nearly 5% of their annual revenue due to this issue [14]. Furthermore, more than 2,700 million electronic transactions are conducted annually in the United States. This growth facilitates financial transactions, but it also increases the number of situations that can be exploited to commit fraud. Ketenci et al. (2021) combined time-frequency analysis with transactional information and CRM data. With this approach, they reduced false positives to 11.85% and were able to better distinguish suspicious entities from those showing no signs of fraud [15].

The rise of digital financial services has not only expanded access but has also changed the rules of the game when it comes to attacks. Systems that operate with fixed rules and predefined limits can still respond effectively to known threats, but they fall short when new or unexpected behaviors emerge. This is where many traditional approaches fall short: they fail to detect in a timely manner those unusual relationships between transactions or the subtle changes in attackers' behavior[16].

Faced with this gap, machine learning began to be seen as a more promising alternative, precisely because it learns from past events. By processing vast amounts of transactions, these models can flag movements that deviate from the norm. And when the information is fluid, or the relationships do not follow a clear logic, probabilistic methods are often the best fit[17]. For his part, Sørensen (2025) analyzes how artificial intelligence and machine learning help uncover signs of fraud that would go unnoticed in a manual review. But he also mentions the problems that remain unresolved, such as false positives and the opacity of some models—two fundamental issues if we are to move toward real-time detection systems that are reliable and, above all, can be clearly explained in the financial sector [18].

The article by Kulkarni et al. (2025) aligns well with what we're seeing here, because it delves deeply into the use of generative AI—VAEs and GANs, to be specific—applied to fraud detection. What's most striking is that these models not only detect very subtle anomalies but can also generate synthetic data without leaking private information. The idea is to build systems that instantly detect fraud in digital payments, without compromising either reliability or the ability to provide clear explanations of how they work [19]. And when we turn to deep learning, the landscape changes because it allows us to analyze transaction sequences at a level of detail that was previously impossible. Networks such as CNNs, LSTMs, autoencoders, and transformers have demonstrated their ability to uncover connections that go completely unnoticed by simpler methods. In fact, in the studies we've reviewed, these architectures have yielded excellent results, both in terms of accuracy and the ability to anticipate future events [20].

But if there's one thing that's proving particularly interesting, it's hybrid models—those that combine the best of traditional machine learning with deep learning. For example, integrating the One-Class SVM with a probabilistic approach has yielded remarkable results: it achieves extremely high accuracy and recall in fraud detection, and best of all, it reduces those false positives. Furthermore, the models become more versatile and adapt better to different scenarios [21].

Finally, speed is key. There's no point in having a system that detects fraud if, by the time it does, it's already too late. That's why the ability to identify suspicious transactions in real time has become essential in digital payments. What researchers have found is that combining models such as transformers, LOF, and Random Forest yields results that excel in precision, recall, and F1-score, demonstrating their enormous potential when applied to complex and fast-paced financial environments [22]. In the words of experts in the field, integrating these approaches is not just a technical improvement: it is a real contribution to the security and efficiency of the digital financial ecosystem as a whole [23].

II. METHODOLOGY

This research is being conducted in accordance with the Systematic Literature Review (SLR) format.

A. PICOC Formulation

The PICOC method was used to determine the scope of the research (Table I).

TABLE I. PICOC COMPONENTS

Component	Description
P (Problem / Population)	Financial transactions suspected of fraud in digital payment systems.
I (Intervention)	Deep learning algorithms with explainability layers.
C (Comparison)	"Black box" deep learning models (without XAI)
O (Results)	Improved interpretability of alerts without significantly compromising accuracy or speed
C (Context)	Fraud prevention departments that require explanations to make operational or regulatory decisions

B. Formulating Questions

After identifying the components of the PICOC method, the general research question (RQ) for this systematic review was formulated:

"What are the deep learning algorithms with Explainable Artificial Intelligence (XAI) applied to the detection of financial fraud in digital payment systems, compared to traditional black-box models, in terms of interpretability, accuracy, and speed?"

Based on this general question, specific questions were formulated (Table II).

TABLE II. PICOC QUESTIONS

RQ1	What machine learning algorithms have been used to detect financial fraud and anomalies in digital payment systems?
RQ2	What deep learning algorithms have been used to detect financial fraud and anomalies in digital payment systems?
RQ3	What machine learning and deep learning models have been used to detect financial fraud in digital payment systems, and to what extent do they outperform or are they outperformed by traditional methods (rule-based and statistical) in terms of effectiveness?
RQ4	What improvements in the interpretability of alerts—while maintaining high levels of precision, recall, F1-score, and a low false positive rate—have been used for fraud detection in digital payment systems?
RQ5	What evidence does the literature provide regarding the effectiveness of real-time models using XAI techniques to improve interpretability without significantly compromising fraud detection performance?

C. Identification of Keywords

The purpose of identifying keywords corresponding to each component of the PICO-C method is to create effective search queries for databases (Table III).

TABLE III. KEYWORDS WITH BOOLEAN OPERATORS

P	"financial transactions" OR "digital payments" OR "payment systems" OR fintech OR banking OR "financial data"
I	"machine learning" OR "deep learning" OR "artificial intelligence" OR "neural networks" OR autoencoder
C	"rule-based" OR "statistical methods" OR "traditional methods" OR "baseline models"
O	"fraud detection" OR "financial fraud" OR "anomaly detection" OR "outlier detection"
C	"real-time" OR "online detection" OR "streaming data" OR "transaction monitoring" OR "explainable AI" OR XAI OR "interpretable machine learning"

D. PICOC Search Formula

Once we had identified the components of the PICOC question and the keywords, we proceeded to develop the search formula that would allow us to find the most relevant literature for this systematic review. To do so, we combined the terms from each component using the Boolean operators "OR" to combine the keywords and "AND" to link the components (Table IV).

TABLE IV. SEARCH EQUATION

Search equation	
Scopus	(TITLE-ABS-KEY ("financial transactions" OR "digital payments" OR "payment systems" OR fintech OR banking OR "financial data") AND TITLE-ABS-KEY ("machine learning" OR "deep learning" OR "artificial intelligence" OR "neural networks" OR autoencoder) AND TITLE-ABS-KEY ("rule-based" OR "statistical methods" OR "traditional methods" OR "baseline models") AND TITLE-ABS-KEY ("fraud detection" OR "financial fraud" OR "anomaly detection" OR "outlier detection") AND TITLE-ABS-KEY ("real-time" OR "online detection" OR "streaming data" OR "transaction monitoring" OR "explainable AI" OR XAI OR "interpretable machine learning")) AND (LIMIT-TO (OA, "all")) AND (LIMIT-TO (LANGUAGE, "English")) AND PUBYEAR > 2020 AND PUBYEAR < 2025
Web of Sciencie	((((((TS=("financial transactions" OR "digital payments" OR "payment systems" OR fintech OR banking OR "financial data")) AND TS=("machine learning" OR "deep learning" OR "artificial intelligence" OR "neural networks" OR autoencoder)) AND TS=("rule-based" OR "statistical methods" OR "traditional methods" OR "baseline models")) AND TS=("fraud detection" OR "financial fraud" OR "anomaly detection" OR "outlier detection")) AND TS=("real-time" OR "online detection" OR "streaming data" OR "transaction monitoring" OR "explainable AI" OR XAI OR "interpretable machine learning")) AND DT=(Article) AND LA=(English)) AND PY=(2021-2026)
Sciencie Direct	("digital payments" OR fintech) AND ("machine learning" OR "deep learning") AND ("fraud detection" OR "anomaly detection") AND ("real-time" OR XAI OR "interpretable machine learning")

E. Inclusion and Exclusion Criteria

The selection of the studies was not arbitrary: specific inclusion and exclusion criteria were applied to ensure their relevance and methodological soundness (Table V).

TABLE V. INCLUSION AND EXCLUSION CRITERIA (PICOC)

Component	Inclusion criteria	Exclusion criteria
Population / Problem	We included research on fraudulent financial transactions in digital payment systems—banks, fintech companies, payment processors, card issuers, digital wallets, and payment gateways—as well as real or synthetic transaction datasets that replicated the financial environment and its characteristic class imbalance.	Studies on non-financial fraud (healthcare, academia, telecommunications, network intrusions) were excluded, unless they justified their transfer to the financial domain. Research on financial transactions not linked to digital payment systems was also excluded.
Scope	Articles that propose, implement, or evaluate deep learning algorithms with explicit Explainable Artificial Intelligence (XAI) mechanisms, such as attention layers, SHAP, LIME, Grad-CAM, gradient-based models, or interpretable architectures.	Studies that used only classical machine learning without deep learning, deep learning models without explainability (black-box models), and studies where XAI was mentioned only as future work without actual development were excluded.
Comparison	Studies that compare explainable models with baseline or reference models—such as neural networks without XAI, Random Forest, XGBoost, SVM, rule-based methods, or traditional statistical approaches—using equivalent experimental conditions.	Research that evaluates only the proposed model without a baseline, or comparisons conducted using different datasets, different preprocessing methods, or non-reproducible metrics.
Results	Preference was given to studies that provided specific performance and	Qualitative studies without accuracy or speed metrics were excluded, as

	interpretability metrics—such as precision, recall, F1-score, ROC-AUC, PR-AUC, latency, throughput, accuracy, or stability—as well as those that included validation by users in the financial sector.	were those using indicators unsuitable for imbalanced data and those with severe performance degradation not justified by their methodology.
Context	The review prioritized studies on fraud detection systems that need to explain operational and regulatory decisions, and included approaches involving online detection, continuous monitoring, and auditing.	Academic studies without real-world validation were excluded, as were batch-processing approaches that ignore latency or fail to explain decisions to regulators.
General Criteria	Articles from databases such as Scopus, Web of Science, and ScienceDirect, preferably in English, published between 2021 and 2026, with full text available and aligned with the review's PICOC question.	We excluded theses, unreviewed technical reports, editorials, unverified documents, duplicates, articles without a DOI or full text, and publications outside the established time frame, language, or scope.

F. PRISMA Statement

PRISMA helped organize the review from the outset. Using these guidelines, we documented how articles were searched for, which ones were excluded, and which ones were included in the final analysis. It also allowed us to present the selection process in a more structured way, so that readers can understand the source of the included studies. According to Page et al.[24], the 2020 version of PRISMA updated the 2009 guidelines to incorporate technological and methodological advances in the search for and analysis of evidence, resulting in a more comprehensive and organized report. Its ability to improve the quality and credibility of publications has led many journals and scientific organizations to adopt it. [24]

According to this study, the PRISMA statement was essential for systematically organizing the process of selecting and analyzing research related to wearable sensors, the Internet of Things (IoT), and artificial intelligence used in mental health monitoring. This guideline was used to establish clear criteria during the selection, exclusion, and final review stages of the articles, ensuring the study's methodological consistency. According to Hutton et al. [25], PRISMA promotes the understanding and comparison of results by encouraging a clear presentation of the techniques used. Thus, it supports transparency, strengthens confidence in the results, and enables future research to replicate and expand upon the findings with greater reliability.

G. PRISMA Process

The PRISMA process was used to review this research. A total of 373 articles were identified: 51 from Scopus, 15 from Web of Science, and 287 from ScienceDirect. After removing duplicates and applying the inclusion and exclusion criteria, the remaining studies were evaluated by reviewing their titles, abstracts, and full texts (Fig. 1).

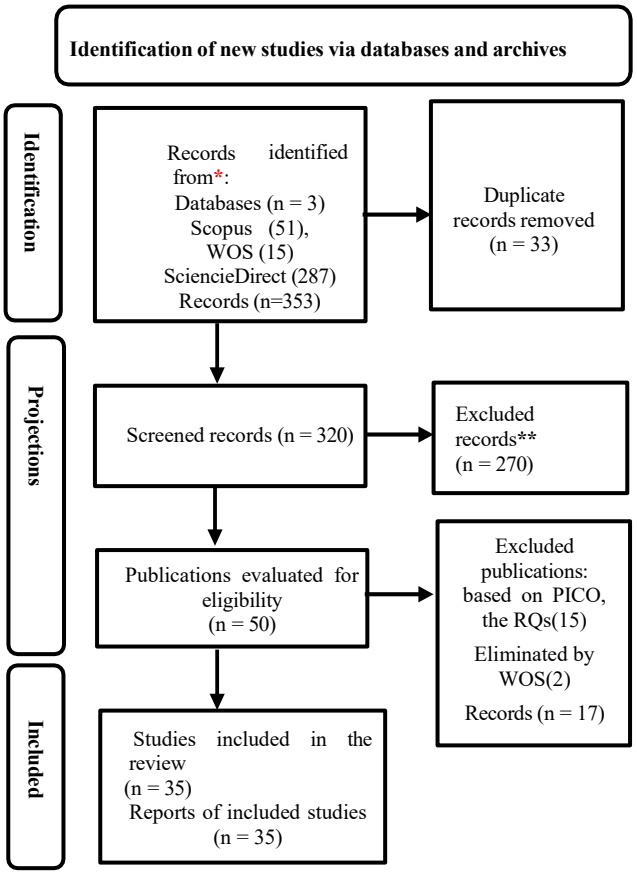


Fig. 1. PRISMA process diagram.

H. Assessment of Methodological Quality

To assess the methodological quality of the 35 studies included in this review, we followed the adapted guidelines of the Joanna Briggs Institute (JBI).[25] Seven criteria were considered: 1) adequate characterization of the dataset (source, size, imbalance, delayed feedback); 2) appropriateness of the data and preprocessing; 3) clear explanation of the algorithm's methodology and the XAI mechanism; 4) rigorous validation (time-series analysis, confidence intervals, statistical tests); 5) use of appropriate metrics for imbalanced data (F1, PR-AUC, latency); 6) explicit discussion of limitations; and 7) quantitative assessment of interpretability. Each of the defined criteria was evaluated on a binary scale—that is, using 0 and 1—which allowed the studies to be classified according to their quality.

Weighted synthesis: Findings from high-quality studies will be given greater weight in the conclusions, while those from moderate-quality studies will be used as a supplement.

I. Bibliometrix

Bibliometrix is an open-source R package, developed by Aria and Cuccurullo (2017), that enables comprehensive bibliometric analyses, including the import of data from databases such as Scopus and Web of Science, as well as the construction of co-citation, collaboration, and keyword co-occurrence networks [26].

One of its most highly valued features is Biblioshiny, its graphical interface based on the Shiny web application, which

allows researchers to use all of its functions without needing prior knowledge of R programming. This combination of analytical power, flexibility, and accessibility has made Bibliometrix one of the most comprehensive and popular tools for the quantitative analysis of scientific output.[27]

III. RESULTS

Following the PRISMA process, 33 articles remained for review. These articles presented different approaches to addressing fraud in digital payments. Several articles utilized machine learning and deep learning. Other studies tested graph networks or incorporated XAI techniques. The applications were varied, although most were related to credit cards, digital banking, blockchain, and money laundering.

The comparison was based on the type of fraud studied, the algorithms used, and the results reported in each article. We also examined how the studies handled the imbalance between normal and fraudulent transactions. In the studies that included explainability, we considered the use of tools such as SHAP and LIME.

The methodological assessment revealed differences among the articles. Of the 33 selected articles, 24 were considered high quality, 5 were of moderate quality, and 2 were of low quality. In the highest-rated studies, it was common to find real-world or publicly available data, comparisons between various models, and results expressed in terms of precision, recall, and F1-score.

When reviewing the limitations, we found that several studies still rely on synthetic data or tests conducted under controlled conditions. In other cases, the time it takes for the model to respond is not reported in sufficient detail. We also found proposals with good results but with little explanation as to why a transaction was considered suspicious.

A. Comparative Overview of Reviewed Studies

1) ML/DL Models and metrics: The results of the reviewed articles are shown in Table VI, which lists the reference, the types of financial fraud in digital payment systems, and the applied ML and DL algorithm models.

TABLE VI. FRAUD TYPES, ML/DL MODELS, AND METRICS

Reference	Types of Financial Fraud in Digital Payment Systems	ML and DL Models	Metrics
Ampumuza, D. et al.[28]	Mobile payment fraud, phishing, card fraud, cryptocurrency scams, identity theft, account takeover	Random Forest, XGBoost, Neural Networks, clustering (K-means), autoencoders, hybrid models (reglas + ML)	No reporta métricas cuantitativas (accuracy, F1, recall, AUC, FPR)
Vazallo-Bautista, J. et al.[29]	CNP, skimming, unauthorized digital payments, account takeover, fake accounts, money laundering using money mule accounts, card fraud (cloning and CNP), phishing, identity theft, suspicious transactions, fraudulent digital payments	Random Forest, XGBoost, Isolation Forest, Autoencoders, LSTM/GRU, CNN, GNN, LightGBM, CatBoost, Hoeffding Tree, Adaptive Random Forest, RNN, GRU, GAN, One-Class SVM, KNN, Logistic Regression, DT, AdaBoost, Stacking Ensemble, CNN-LSTM with Attention	Accuracy (94-99%), Precision (>95%), Recall (90-98%), F1-Score (0.90-0.999), AUC-ROC (>0.98), MCC, False Positive Rate (reducción >40%), Latency (<100 ms), throughput (tx/s)
Alatawi, M. N. et al.[30]	Credit card fraud (CNP, skimming, unauthorized digital payments, online fraud, IoT-based transactions)	Logistic Regression, Random Forest, Gradient Boosting Machine (GBM), Multilayer Perceptron (Neural Networks)	Accuracy (RF: 99.999%, GBM: 99.9865%, MLP: 99.876%, LR: 95.87%), Precision (RF: 1.0, GBM: 0.9995, MLP: 0.9970, LR: 0.8911), Recall (RF: 0.9999, GBM: 0.9990, MLP: 0.9888, LR: 0.6000), F1-Score (RF: 0.9999, GBM: 0.9992, MLP: 0.9929, LR: 0.7171)
Abd-Ellatif, L. et al.[31]	Credit card fraud, identity theft, phishing, unauthorized fund transfers	ATAD-Net (CNN + LSTM), CNN, RNN	Accuracy: ATAD-Net 98.65%, Precision: 97.12%, Recall: 96.74%, F1-Score: 96.93%, Average latency: 8.2ms por transacción, Throughput: reportado en Figura 8 (TPS), True Positives, False Positives, True Negatives, False Negatives (confusion matrix Figura 2)
Al-Mustaqbal University et al. [32]	Credit card fraud, unauthorized online transactions, electronic financial crimes	Decision Tree, Logistic Regression, Naive Bayes, Random Forest, ANN, SVM, KNN, CNN, LSTM, GAN, Ensemble methods (Bagging, Boosting)	Accuracy (SMOTE: 98.1%, undersampling: 94.9%, oversampling: 95.3%, GAN: 99.9%), Precision (SMOTE: 99.7%, undersampling: 98.8%, oversampling: 99.7%, GAN: 99.9%), Recall (SMOTE: 98.1%, undersampling: 94.9%, oversampling: 95.2%, GAN: 99.9%), F1-Score (SMOTE: 98.1%, undersampling: 94.7%, oversampling: 95.1%, GAN: 99.9%), ROC-AUC (SMOTE: 96%, undersampling: 95%, oversampling: 95.5%, GAN: 100%)
Leong, W. Y. et al.[33]	Identity theft, payment fraud, account takeover, credit card fraud, phishing	Logistic Regression, Decision Trees, Random Forest, Neural Networks, CNN, RNN, Isolation Forest, Autoencoders	Accuracy (LR: ~91%, RF: ~95%, SVM: ~90%, LSTM: ~97%, Autoencoder: ~82%), Precision, Recall, F1-Score,

Reference	Types of Financial Fraud in Digital Payment Systems	ML and DL Models	Metrics
			AUC (LSTM: 0.98, RF: 0.96), curvas precision-recall, matrices de confusión
El Youbi, R. et al.[34]	Electronic payment fraud, credit card fraud, unauthorized transactions, phishing	DNN, LSTM RNN, Logistic Regression, KNN, Random Forest, CNN, Deep Reinforcement Learning + ANN	Accuracy: 99.72%, AUC: 97.43%, Precision (fraud): 96.0%, Recall (fraud): 93.8%, F1-Score (fraud): 94.9%, Precision (no fraud): 99.8%, Recall (no fraud): 99.9%, F1 (no fraud): 99.8%, ROC-AUC: 0.999, PR-AUC: 0.913, Average Precision: 0.913, Confusion Matrix: TN=14,855, FP=17, FN=1, TP=127
Al Lawati, H. M. R. et al. [35]	Card fraud (skimming, CNP, counterfeit cards, loss/theft), payment fraud, phishing, collusion with merchants	CNN, ADASYN + Mutual Information + SelectKBest, EDDM, ADWIN, Random Forest, XGBoost, LightGBM, SVM, KNN, Decision Tree, J48, K-means, GRU	Accuracy (Credit Card: 100%, Real: 100%, Spam: 99.34%), Precision (Credit Card: 100%, Real: 100%, Spam: 100%), Recall (Credit Card: 100%, Real: 100%, Spam: 98.69%), F1-Score (Credit Card: 100%, Real: 100%, Spam: 99.34%), ROC-AUC (Credit Card: 100%, Real: 100%, Spam: 99.34%), Drift Detection Rate (Credit Card: 100%, Real: 100%, Spam: 99.34%)
Pavlova, O. et al.[36]	Money laundering, drop accounts, cryptocurrency fraud (Bitcoin)	GNN, GCN, DNN, Random Forest, Logistic Regression, MLP, CNN, KNN, LASSO, Gradient Boosting, Elastic Net	Accuracy: 93%, Recall: 88%, F1-Score: 90%, Precision-Recall AUC: valores altos según la curva (Figura 5), Confusion Matrix con TP, TN, FP, FN (Figura 4). La Precision-Recall curve muestra buen balance pero Precision más baja que Recall
Iqbal, S. et al.[37]	Credit card fraud, online transaction fraud	Random Forest, XGBoost, SHAP, LIME	RF: Accuracy 0.9996, Precision 0.8909, Recall 0.9423, F1 0.9159, ROC-AUC 0.9979, CV-F1 0.9998. XGB: Accuracy 0.9995, Precision 0.8776, Recall 0.9556, F1 0.9149, ROC-AUC 0.9997, CV-F1 0.9998
Renuga Devi, R. et al.[38]	Credit card fraud, real-time fraudulent transactions, clustered fraud patterns (coordinated attacks or organized fraud), detection of suspicious communities (cardholders, merchants, and compromised devices)	GNN (Graph Attention Networks), RL (Deep Q-Network), GAN, RNN (GRU, LSTM), SVM, Random Forest, XGBoost, Autoencoders, Restricted Boltzmann Machines, CART	AUROC 0.872, AP 0.683, Recall 0.85, Precision 0.83, F1 0.839, Balanced Accuracy 0.916, MCC 0.54, FPR 0.08, Latency 42ms/batch, Throughput 12,000 tx/s
Černevičienė, J. & Kabašinskas, A.[39]	General fraud detection (including credit card fraud, detection of suspicious transactions, electronic payment fraud, and P2P loan fraud)	ANN, XGBoost, Random Forest, SHAP, LIME, LightGBM, CNN, LSTM, Autoencoders, Gradient Boosting, Decision Trees, Logistic Regression, KNN, SVM, frameworks multi-enfoque	SLR cualitativa. 138 artículos, 397 autores, 80% publicados 2020-2022, China (33), Italia (17), Alemania (11), coautoría internacional 10%
Prashanth, B. S. et al.[40]	Banking transaction fraud, digital banking fraud, detection of suspicious transactions in electronic payment systems	TabNet, DNN, GRU, LSTM, CNN1D, SMOTE	Accuracy 97.39%, ROC-AUC 0.9739, Fold AUCs: 0.9736,0.9733,0.9747. LSTM AUC 0.5118, GRU AUC 0.517, CNN1D AUC 0.50, DNN AUC 0.5
Kungu, C. O. et al.[41]	Money laundering, cross-border payment fraud, suspicious transactions	VAE, GNN, One-Class SVM, Isolation Forest, LOF, SHAP	Precision@5%: OCSVM 99.63%, IF 84.92%, LOF 89.76%. AP: OCSVM 0.8478, IF 0.6742, LOF 0.8019. ROC-AUC: OCSVM 0.8388, IF 0.7826, LOF 0.8459. Throughput: 1000 TPS
Karnavou, E. et al.[42]	Credit card fraud (banking transaction fraud)	Isolation Forest, One-Class SVM, Autoencoder, SHAP	Isolation Forest (Merchants DE FS): TP 59%, FP 5%, Acc 93%, F1 15%, AUC 80%, Precision 12%. Persons: TP 48%, F1 20%, AUC 70%, Precision 17%. Autoencoder Persons: TP 85%, F1 20%, AUC 88%, Precision 11%
Gupta, R. K. et al.[43]	Credit Card Fraud	SVM, KNN, ELM, Stacking Ensemble (Gradient Boosting), Autoencoder.	Stacking Ensemble: Acc 99.95%, Prec 99.93%, Rec 99.97%, F1 99.95%, AUC 1.00, FPR 0.02%. SVM: Acc 97.33%, Prec 98.69%, Rec 96.03%, F1 97.34%, AUC 0.98. KNN: Acc 99.87%, Rec 100%, F1 99.87%, AUC 0.99

Reference	Types of Financial Fraud in Digital Payment Systems	ML and DL Models	Metrics
Hasan, M. et al.[44]	Fraud in blockchain transactions (Bitcoin), fraudulent cryptocurrency transactions	1D CNN, XGBoost, Random Forest, Decision Tree, Gradient Boosting, AdaBoost, SHAP	1D CNN+SMOTE (early stop): Acc 0.98, TPR 0.36, FPR 0.02. 1D CNN+ADASYN: Acc 0.99, TPR 0.27, FPR 0.005. 1D CNN+SMOTEENN: Acc 0.99, TPR 0.27, FPR 0.0008
Chen, Y. et al.[45]	Credit card fraud, bank fraud, insurance fraud, cryptocurrency/blockchain fraud, money laundering	CNN, LSTM, Transformer, GNN, GAN, VAE, Autoencoder-LSTM, CatBoost-DNN, Random Forest, XGBoost, LightGBM.	108 estudios. LSTM 25.7%, MLP 20%, CNN 15.7%, RNN 14.3%, Transformers 11.4%. Credit card sector más estudiado. 92/108 reportan desbalanceo. Métricas: accuracy, precision, recall, F1, AUC-ROC, AUC-PR, FPR, FNR, cost metrics
Murugaraj, G. et al.[46]	Credit card fraud, fraud detection, anti-money laundering (AML)	QSVM, QNN, QAOA, Quantum Clustering, Quantum GNN, Quantum Amplitude Estimation, Quantum Boltzmann Machines, Hybrid Quantum-Classical models	QSVC (ZFeatureMap): F1=0.98 (BankSim). QNN: 97.63% accuracy. QGNN: 94.5% accuracy. QAOA: speedups 10-1000x. QAE: quadratic speedup. QADQN: 75.37% returns
Mia, M. S. et al.[47]	Credit card fraud	Isolation Forest, CNN, Autoencoder, XGBoost, Decision Tree, KNN, Logistic Regression, LSTM, Random Forest, SVM, VQC (Quantum), SMOTE, AutoEncoder + Bagging (RF) + Boosting (XGBoost)	Proposed Model-1: Acc 100%, Prec 1.00, Rec 1.00, F1 1.00, AUC 0.9998. XGBoost: Acc 100%, Prec 0.81, Rec 0.93, F1 0.86, AUC 0.9841. RF: Acc 100%, Prec 0.94, Rec 0.91, F1 0.92, AUC 0.9684. VQC: Acc 86%, Prec 1.00, Rec 0.72, F1 0.84
Uddin, N. et al.[48]	Financial fraud in general (credit card fraud, money laundering, identity theft, digital payment fraud)	Decision Trees, Random Forest, XGBoost, Neural Networks, Isolation Forest, Hybrid Models (reglas + ML), Explainable AI (XAI)	Accuracy puede ser engañosa en datasets desbalanceados. Se recomienda Precision (>70%), Recall, F1-Score, AUC-ROC. Pérdidas EE.UU.: $12.5B (2024)
Usman, A. et al.[49]	Money laundering (AML), fraudulent transactions in payment systems	Graph Convolutional Networks, Semi-supervised learning, PageRank, Word2Vec, Doc2Vec, TransE, ComplEX	Accuracy 79.36% promedio (79.96% max). RF F1 0.619, NN 0.479, DT 0.423. Dataset: 100K nodos, 5.4M aristas
Marazqah Btoush, E. et al.[50]	Credit card fraud, cyber fraud in banking transactions, online payment fraud	LR, RF, SVM, DT, NB, KNN, ANN, CNN, RNN, LSTM, GRU, XGBoost, LightGBM, CatBoost, AdaBoost, GAN, Autoencoder, Isolation Forest, LOF, HMM, K-means, SMOTE, ADASYN, RUS, Tomek	Revisión sistemática; no aplica una métrica única por modelo. Sintetiza resultados y tendencias de ML/DL en fraude con tarjetas.
Ahmad, H. et al.[51]	Credit card fraud	ANN, XGBoost, RF, LR, SVC, GB, Tomek Links (undersampling)	Accuracy ANN 99.36% (D1), 99.36% (D2); Recall ANN 99.41% (D1), 98.92% (D2); Precision ANN 99.73% (D1), 98.73% (D2); F1 ANN 99.36% (D1), 99.37% (D2); G-Mean ANN 99.36% (D1), 99.21% (D2). RF Accuracy 98.57% (D1), GB 98.57% (D1), SVC 97.61% (D1)
Miao, Z. et al.[52]	Financial fraud (credit card fraud, e-commerce fraud, electronic payment fraud, cryptocurrency fraud)	Transformer, LOF, Random Forest, XGBoost, LightGBM, LSTM, Isolation Forest	Accuracy 95.47%, Precision 94.32%, Recall 96.58%, F1 95.44%, AUC 97.12%, MCC 92.76%, Specificity 95.13%. Transformer: CCFD Acc 89.63%, EPF Acc 96.23%, FTTS Acc 91.63%, VCT Acc 92.89%. LOF: CCFD Acc 96.2%, EPF Acc 95.48%, FTTS Acc 96.22%, VCT Acc 92.39%
Khan A. et al.[53]	Credit card fraud	LSTM-RNN, Naive Bayes, SVM, ANN	LSTM-RNN: Accuracy 100%, Precision 1.00, Recall 1.00, F1-Score 1.00, Sensitivity 1.00. ANN: Accuracy 91.85%, Precision 92.6%, F1 71.6%. SVM: Accuracy 86.56%, Precision 68.8%. Naive Bayes: Accuracy 69.54%, Precision 71.6%
Venkata Lakshmi Narayana Gorle. et al.[54]	Debit and credit card fraud (unauthorized purchases), online banking fraud, document forgery, false financing, scams, fake accounts, financial statement fraud (alteration of accounting records), identity theft, phishing, mobile payment	Deep Learning Models: AM-1DCNN (Adaptive Multiscale 1D Convolutional Neural Network), A-CNet (Adaptive Capsule Network). Machine Learning Models (feature extraction): PCA (Principal Component Analysis), t-SNE (t-distributed Stochastic	IWPA-AM-1DCNN: Acc 94.18%, Prec 94.15%, Rec 94.23%, F1 93.86%, Spec 94.17%, MCC 88.36%, FPR 5.83%. IWPA-A-CNet: Acc 93.76%, Prec 93.62%, F1 93.69%, MCC 87.52%, FPR 6.25%.

Reference	Types of Financial Fraud in Digital Payment Systems	ML and DL Models	Metrics
	fraud, cryptocurrency scams, account takeover.	Neighbor Embedding), SVM (Support Vector Machine). Optimization algorithm: IWPA (Improved Waterwheel Plant Algorithm). Compared ensemble techniques: XGBoost, IMEML, HEML, RILHHO.	Validación externa: Dataset2 94.27%, Dataset3 93.91%, Dataset4 95.87%, Dataset5 94.96%.
Wang, Y., Zheng, Q.[54]	Blockchain/cryptocurrency fraud, money laundering, phishing, scams on dark markets (Silk Road, Hydra), illicit Bitcoin/Ethereum transactions, identity theft, account takeover.	DL: CoSemiGNN (dynamisches Graph-Neuralnetzwerk mit semi-überwachter Ko-Assoziation), EvolveGCN, LRGCN, GConvGRU, GC-LSTM, DCRNN, A3T-GCN, AGCRN, DyGrEncoder. ML: GCN, GAT, Random Forest, XGBoost, logistische Regression, Entscheidungsbaum, SVC, KNN, MLP, AdaBoost, Struc2Vec. Techniques: Halbüberwachtes Lernen (SelfTrain, sim-kernelKmeans), LSTM, GRU, Multi-Head-Attention.	CoSemiGNN-E (mejor): Acc 98.47%, Illegal-F1 80.16%, Legal-F1 99.17%, Micro-F1 89.66%, Illegal-Prec 96.43%, Illegal-Rec 71.11%. CoSemiGNN-V: Acc 98.14%, Illegal-F1 74.46%, Legal-F1 98.99%, Micro-F1 86.73%, Illegal-Prec 77.57%, Illegal-Rec 73.02%. Comparativa: Superior a GCN (Illegal-F1 38.45%), GAT (36.74%), EvolveGCN-O (43.83%), DCRNN (35.78%).
Lebichot, B. et al.[55]	Credit card fraud (unauthorized transactions), e-commerce fraud, digital payment scams.	DL: NN (dense multilayer network), EWC (Elastic Weight Consolidation), IMM (Incremental Moment Matching), GenR (Generative Replay with CGAN), Frz (Frozen Network). ML: Batch, Retrain, Incremental, Experience Replay (ExpR), Negative Correlation Ensemble (NCE). Techniques: Ensemble learning, rehearsal, regularization, pseudo-rehearsal, weight transfer.	FrzE (Frozen Network Ensemble - mejor): Pr@100 25.36% ± 6.49%, AUPRC 11.28% ± 2.90%. Comparativa: - NCE (Ens): Pr@100 26.24%, AUPRC 11.66% - Incr (Ens): Pr@100 24.28%, AUPRC 11.20% - EWC: Pr@100 24.19%, AUPRC 10.59% - Batch: Pr@100 20.54%, AUPRC 6.97% Tiempos: FrzE 89 min, NCE 126 min, EWC 657 min.
Chang, V. et al.[56]	Credit/debit card fraud (unauthorized transactions), digital payment fraud, e-commerce, online banking.	ML (supervised): Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF). DL (no supervisado): Autoencoder (AE). Techniques: PCA (feature extraction), SMOTE (oversampling), NearMiss (undersampling), GridSearchCV, 5-fold cross-validation.	RF + PCA + NearMiss (mejor): Acc 98%, AUROC 0.976, Avg Prec 0.927, Sens 0.83, Spec 1.00. LR + NearMiss (mejor precisión): AUROC 0.972, Avg Prec 0.928. Autoencoder (5b - mejor AE): Acc 96%, AUROC 0.947, Avg Prec 0.809, Sens 0.81, Spec 0.98. Comparativa supervisada (PCA): - LR: AUROC 0.971, Avg Prec 0.927 - KNN: AUROC 0.933, Avg Prec 0.865 - DT: AUROC 0.900, Avg Prec 0.858 - RF: AUROC 0.960, Avg Prec 0.921
Kolli, C.S. et al. [57]	UPI (Unified Payment Interface) transaction fraud, phishing, SIM swap fraud, social engineering, fake UPI apps, payment interface cloning, identity theft, unauthorized access to bank accounts.	Deep Learning Models: SNN (Spiking Neural Network) optimized with SCWO.Modelos de Machine Learning: RF (Random Forest), ML (Machine Learning), XFL (Explainable Federated Learning). Techniques: FL (Federated Learning), Borderline-SMOTE (oversampling), SCWO (Snow Carpet Weaver Optimization—a combination of SAO and CWO), Comprehensive Normalization.	SCWO-SNN (propuesto): Loss 0.119, NMSE 0.197, NRMSE 0.443, Prec 91.400%, Rec 91.275%, F1 91.448%. Comparativa (Training data=90%): - RF: Prec 77.626%, Rec 78.415%, F1 77.848% - XFL: Prec 79.966%, Rec 79.636%, F1 79.801% - Federated SDT: Prec 80.349%, Rec 80.389%, F1 80.663% - ML: Prec 82.114%, Rec 82.996%, F1 82.851%
Backiyalakshmi, V., & Umadevi, B.[58]	Fraud in the banking sector (bank accounts), credit/debit card fraud, identity theft, unauthorized transactions, insurance fraud, accounting fraud, and fraud involving bank card registrations.	Deep Learning Models: DCL-MDA (Dilated ConvLSTM + Multi-scale Dense Attention). Machine Learning Models: GAN, DRL, GBDT, GNN, ABC, HLBO, POA, MRSA. Techniques: PCA, t-SNE, Statistical Features, MRV-RSA (Modified Random Value Reptile Search Algorithm), weighted feature fusion.	DCL-MDA + MRV-RSA (mejor modelo): - Dataset 1: Acc 93.86%, Prec 2.57%, True fraud 3,200, False fraud 26,684,700, Error 6.34%. - Dataset 2: Acc 97.15%, Prec 18.91%, True fraud 300, False fraud 965,000, Error 4.03%.

IV. Research Questions (RQs) and Components

The following table summarizes the most relevant references from the 33 studies screened for each research question. Priority was given to articles that directly address each research question, to avoid an excessive number of references per row (Table VII).

TABLE VII. Research Questions, Titles, and References

Question Types	Question Title	References
RQ1	Machine learning algorithms used to detect financial fraud and anomalies in digital payment systems.	[29],[30], [35], [37], [41], [43], [14]
RQ2	Deep learning algorithms used to detect financial fraud and anomalies in digital payment systems.	[31], [34], [38], [40], [6],[53], [58]
RQ3	Comparison of machine learning and deep learning models with traditional rule-based or statistical methods.	[32], [17], [43], [16], [23],[14]
RQ4	Improvements in the interpretability of alerts while maintaining accuracy, recall, F1-score, and a low false positive rate.	[17], [37], [39],[41], [42], [44], [48]
RQ5	Evidence on real-time models with XAI to improve interpretability without compromising performance.	[29], [31], [35], [38], [41], [52], [57]

A. Bibliometric Research Figures

Bibliometric analysis complements the systematic review by providing a visual overview of the trends, relationships, and clusters present in the selected studies. In this case, the figures show how topics related to real-time anomaly detection, the use of machine learning and deep learning, explainable artificial intelligence, and the prevention of financial fraud in digital payment systems are organized. The interpretation is presented in a formal manner, while striving to maintain clear and natural wording so that it can be integrated directly into the final document.

B. Word Cloud

Fig. 2: In the word cloud, the most prominent terms are financial fraud, machine learning, deep learning, anomaly detection, and digital payments. This aligns with the selected articles, as the vast majority focus on how to identify suspicious transactions using machine learning models.



Fig. 2. Word cloud highlighting the dominant concepts in the field under study.

Concepts related to XAI, financial transactions, and interpretability also appear. Their presence indicates that several studies aim not only to improve detection but also to explain why a transaction was deemed risky. Thus, the figure highlights the most important themes of our review.

C. Scientific Collaboration Network

Fig. 3: The scientific collaboration network exhibits a structure composed of several small groups and a few more prominent collaboration hubs, indicating that scientific output on financial fraud detection using AI is distributed across relatively independent communities. This characteristic is understandable given that the topic encompasses diverse fields such as machine learning, deep learning, cybersecurity, fintech, blockchain, and digital payments. The groups with the highest density suggest the existence of teams engaged in ongoing collaboration and with more established lines of research.

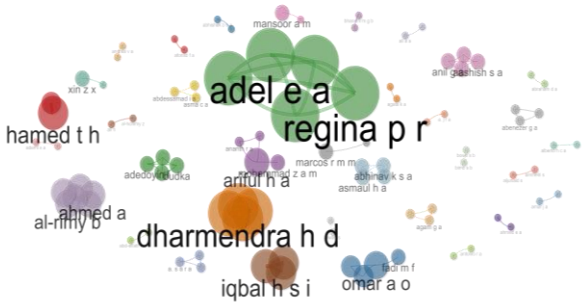


Fig. 3. Scientific collaboration network identified in the analyzed documents.

D. Conceptual Structure Map

Fig. 4: The map identifies three groups of topics. The green section includes support vector machines, SHAP, real-time fraud detection, and card transactions. In these studies, the focus is not only on detecting a suspicious transaction but also on understanding why the model flagged it and how long it takes to do so.

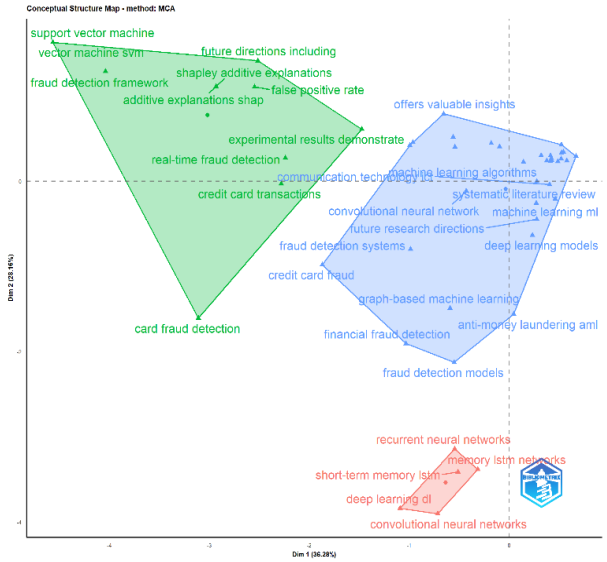


Fig. 4. Concept map generated using multiple correspondence analysis.

The blue area includes machine learning, convolutional networks, graph models, financial fraud, and money laundering. The articles in this group compare different ways of identifying transactions that could be fraudulent.

The red group includes deep learning, recurrent neural networks, LSTM, and CNN. These models are used when fraudulent behavior depends on multiple transactions or relationships that are not easily recognized. Looking at all three groups, the map reveals that the review is primarily divided among explainability, fraud detection, and deep models.

E. Thematic Map

Fig. 5: In the figure, machine learning, artificial intelligence, finance, blockchain, fintech, and fraud detection appear among the most prevalent topics. This is because they recur in several of the analyzed articles. Deep learning and federated learning appear in another section of the map. Meanwhile, document examination, digital twin, metaverse, privacy, and security compliance are less prevalent within the analyzed dataset. A similar pattern is observed for ensemble learning, class imbalance, generative AI, and large language models, which do not yet have the same prominence as the central topics. The position of each group reveals that some research areas already have more studies, while others still appear less frequently.

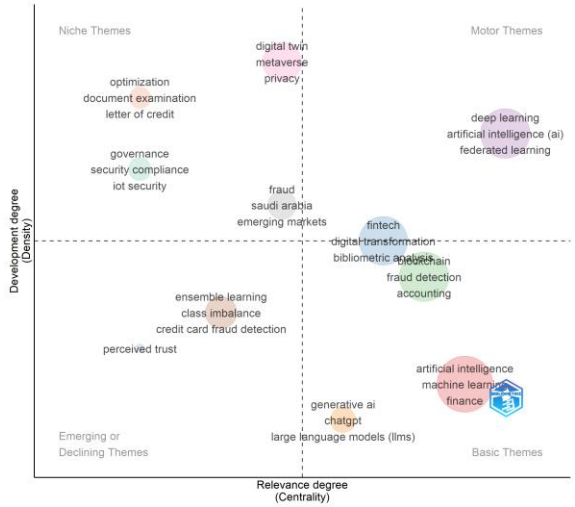


Fig. 5. Thematic map based on the centrality and density of the identified topics. (Source: Prepared by the author based on bibliometric analysis in Bibliometrix.)

V. DISCUSSION

The review shows that rule-based systems are still useful when fraud follows known patterns. The problem arises when those patterns change, or new forms of fraud emerge, because these systems are less adaptable. For this reason, the most recent studies increasingly rely on machine learning and deep learning models. Even so, the results do not allow us to identify a single model as the best in all cases.

For RQ1, Random Forest, XGBoost, Support Vector Machine, Logistic Regression, and Isolation Forest are among the most commonly used algorithms. In several studies, they achieve competitive results, although accuracy may not be very representative when fraudulent transactions are much less frequent than normal ones. In these situations, metrics such as recall, F1-score, ROC-AUC, and the false positive rate allow for a better evaluation of the model's performance.

In RQ2, architectures such as LSTM, CNN, GNN, GRU, and autoencoders were identified. These techniques can analyze more complex relationships and transaction sequences, but they also require more resources and tend to be more difficult to interpret. This can become a limitation when the model is used in a financial environment where an alert must be processed and understood quickly.

RQ3 revealed that traditional methods retain the advantage of being easy to interpret, although they are more difficult to adapt to changes in fraudulent behavior. Mill et al. point out that the increase in the volume of information and the emergence of new forms of fraud make it necessary to supplement these systems with machine learning techniques. However, a more complex model does not necessarily guarantee a better result.

In RQ4, SHAP and LIME make it possible to show which variables carried the most weight in an alert. However, this does not mean that the explanation is necessarily useful to the analyst. Diaz et al. indicate that there is still little empirical evidence on how AML specialists use these explanations in real-world settings[59]. Hosawi and Stone also caution that an unclear explanation can increase cognitive load or lead to overconfidence in the system[60].

Finally, RQ5 shows that there are still limitations to evaluating XAI in real-time fraud detection scenarios. Mill et al. indicate that factors such as false positives, class imbalance, concept drift, and real-time analysis should be considered together [61]. It is not enough to measure the algorithm's speed; it is also necessary to consider how much time the analyst needs to understand the alert and make a decision.

VI. CONCLUSION

Based on the reviewed studies, it was found that machine learning and deep learning can help detect fraud that rule-based systems alone do not always recognize. Machine learning models are generally easier to implement and explain, while deep learning models can identify more complex relationships among transactions.

It also became clear that high accuracy alone does not guarantee a good outcome. In this type of problem, it is important to know how many instances of fraud the model detects, how many false positives it generates, and whether it can respond quickly.

XAI techniques can help explain why a transaction was flagged as suspicious. However, several studies have not demonstrated whether these explanations actually assist the analyst in making a decision. Another limitation is that many studies use public or synthetic data, so it cannot be guaranteed that the same performance will be achieved in a real bank or payment platform.

Based on the review, the combination of machine learning, deep learning, and XAI can be useful for improving fraud detection. However, further testing with real data, larger transaction volumes, and the involvement of financial specialists

is still needed. This would make it possible to verify whether the models perform well outside of laboratory tests and whether their results can be put to practical use.

REFERENCES

[1] D. Ampumuza, C. Katushabe, and M. Tamale, "A systematic review and future directions for AI-driven detection of fraud patterns in SACCO transactions," Front. Artif. Intell., vol. 8, Jan. 2026, doi: 10.3389/frai.2025.1690482.

[2] S. S. and C. S, "Comparative Study on Anomaly-Based Intrusion Detection using Deep Learning Techniques," EAI Endorsed Transactions on Internet of Things, vol. 11, Nov. 2024, doi: 10.4108/eetiot.7178.

[3] B. Fu, Y. Tong, Y. Li, Z. Tang, Z. Shang, and A. Li, "Financial Fraud Anomaly Detection of Listed Companies Based on Probabilistic Perspective Machine Learning Models," Procedia Comput. Sci., vol. 266, pp. 979–986, 2025, doi: 10.1016/j.procs.2025.08.121.

[4] Y. Wang, D. Lian, and E. Chen, "Fraud Detection Based on Credit Review Texts with Dual Channel Memory Networks," Applied Artificial Intelligence, vol. 38, no. 1, Dec. 2024, doi: 10.1080/08839514.2024.2385854.

[5] M. R. Raja, M. A. Hosen, M. F. Kabir, S. Sultana, S. A. Ashraf, and R. Islam, "Detecting and Preventing Money Laundering Using Deep Learning and Graph Analysis," International Journal of Advanced Computer Science and Applications, vol. 16, no. 6, 2025, doi: 10.14569/IJACSA.2025.0160601.

[6] Y. Chen, C. Zhao, Y. Xu, C. Nie, and Y. Zhang, "Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications," Data Science and Management, Aug. 2025, doi: 10.1016/j.dsm.2025.08.002.

[7] Y. Li, B. Fu, Y. Tong, Z. Tang, Z. Shang, and A. Li, "Abnormal Detection of Financial Fraud in Listed Companies Based on Deep Learning," Procedia Comput. Sci., vol. 242, pp. 1402–1409, 2024, doi: 10.1016/j.procs.2024.08.112.

[8] J. Nicholls, A. Kuppa, and N.-A. Le-Khac, "Financial Cybercrime: A Comprehensive Survey of Deep Learning Approaches to Tackle the Evolving Financial Crime Landscape," IEEE Access, vol. 9, pp. 163965–163986, 2021, doi: 10.1109/ACCESS.2021.3134076.

[9] J. Černevičienė and A. Kabašinskas, "Explainable artificial intelligence (XAI) in finance: a systematic literature review," Artif. Intell. Rev., vol. 57, no. 8, p. 216, Jul. 2024, doi: 10.1007/s10462-024-10854-8.

[10] B. Lebichot, W. Siblini, G. M. Paldino, Y.-A. Le Borgne, F. Oblé, and G. Bontempi, "Assessment of catastrophic forgetting in continual credit card fraud detection," Expert Syst. Appl., vol. 249, p. 123445, Sep. 2024, doi: 10.1016/j.eswa.2024.123445.

[11] H. M. R. Al Lawati et al., "An Integrated Preprocessing and Drift Detection Approach With Adaptive Windowing for Fraud Detection in Payment Systems," IEEE Access, vol. 13, pp. 92036–92056, 2025, doi: 10.1109/ACCESS.2025.3569609.

[12] S. K. Aljunaid, S. J. Almheiri, H. Dawood, and M. A. Khan, "Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection," Journal of Risk and Financial Management, vol. 18, no. 4, p. 179, Mar. 2025, doi: 10.3390/jrfm18040179.

[13] V. Backiyalakshmi and B. Umadevi, "MRV-RSA: Developed Modified Random Value Reptile Search Algorithm and Deep Learning based Fraud Detection Model in Banking Sector," Data Knowl. Eng., vol. 163, p. 102557, May 2026, doi: 10.1016/j.datak.2026.102557.

[14] V. Chang, L. M. T. Doan, A. Di Stefano, Z. Sun, and G. Fortino, "Digital payment fraud detection methods in digital ages and Industry 4.0," Computers and Electrical Engineering, vol. 100, p. 107734, May 2022, doi: 10.1016/j.compeleceng.2022.107734.

[15] U. G. Ketenci, T. Kurt, S. Onal, C. Erbil, S. Akturkoglu, and H. S. Ilhan, "A Time-Frequency Based Suspicious Activity Detection for Anti-Money Laundering," IEEE Access, vol. 9, pp. 59957–59967, 2021, doi: 10.1109/ACCESS.2021.3072114.

[16] Md. S. Mia, S. Roy, M. A. Ihsan, S. Hossain, and Md. K. U. Ahamed, "Data-driven financial fraud detection using hybrid artificial and quantum intelligence," BenchCouncil Transactions on Benchmarks, Standards and Evaluations, vol. 5, no. 4, p. 100252, Dec. 2025, doi: 10.1016/j.tbench.2025.100252.

[17] N. K. Narasapuram, S. A. A. Shah, M. F. Shiratuddin, and F. Sohel, "Explainable artificial intelligence models for detecting suspicious bank transactions," International Journal of Machine Learning and Cybernetics, vol. 17, no. 3, p. 111, Mar. 2026, doi: 10.1007/s13042-026-02994-w.

[18] E. Y. Wirawan, M. A. Rahman, and E. Kurniawan, "Artificial Intelligence Implementation in Fraud Detection for Financial Due Diligence Project Financing in Indonesia Company," 2026, pp. 273–284. doi: 10.1007/978-3-032-01133-6_19.

[19] P. Kulkarni, P. Pathak, S. Pillai, and V. Tigga, "Role of Generative AI for Fraud Detection and Prevention," in Generative Artificial Intelligence in Finance, Wiley, 2025, pp. 175–198. doi: 10.1002/9781394271078.ch10.

[20] E. A. L. Marazqah Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran, "A systematic review of literature on credit card cyber fraud detection using machine and deep learning," PeerJ Comput. Sci., vol. 9, p. e1278, Apr. 2023, doi: 10.7717/peerj-cs.1278.

[21] V. L. Narayana Gorle and S. Panigrahi, "An efficient heuristic optimization-based fraudulent activities detection in the financial sector using adaptive machine learning and deep learning system," Expert Syst. Appl., vol. 302, p. 130551, Mar. 2026, doi: 10.1016/j.eswa.2025.130551.

[22] W. Y. Leong, Y. Z. Leong, and W. S. Leong, "Artificial Intelligence-Driven Fraud Detection: Enhancing Security in Digital Age," 2026, pp. 65–78. doi: 10.1007/978-981-95-2113-5_7.

[23] H. Ahmad, E. Rawashdeh, A. AlTawil, and N. Al-Ramah, "EFC-Tomek: An effective undersampling technique for credit card fraud detection," International Journal of Data and Network Science, vol. 9, no. 4, pp. 845–852, 2025, doi: 10.5267/j.ijdns.2025.7.003.

[24] M. J. Page et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," BMJ, p. n71, Mar. 2021, doi: 10.1136/bmj.n71.

[25] A. A. Veroniki et al., "Updating the PRISMA reporting guideline for network meta-analysis: a scoping review," J. Clin. Epidemiol., vol. 188, p. 111985, Dec. 2025, doi: 10.1016/j.jclinepi.2025.111985.

[26] M. G. Belu and A. M. Marinoiu, "AI-Enabled Supply Chain Management: A Bibliometric Analysis Using VOSviewer and RStudio Bibliometrix Software Tools," Sustainability, vol. 17, no. 5, p. 2092, Feb. 2025, doi: 10.3390/su17052092.

[27] M. Rodrigues, A. D. Daniel, and M. Franco, "What is important to know about mumpreneurship? A bibliometric analysis," International Journal of Organizational Analysis, vol. 31, no. 7, pp. 3413–3435, Nov. 2023, doi: 10.1108/IJOA-05-2022-3293.

[28] D. Ampumuza, C. Katushabe, and M. Tamale, "A systematic review and future directions for AI-driven detection of fraud patterns in SACCO transactions," Front. Artif. Intell., vol. 8, Jan. 2026, doi: 10.3389/frai.2025.1690482.

[29] J. Vazallo-Bautista, A. Villalobos-Peña, and J. Soria-Quijaite, "Intelligent Systems, Machine Learning, and Deep Learning Algorithms for Detecting Banking Fraud: A Review," International Journal of Advanced Computer Science and Applications, vol. 16, no. 12, 2025, doi: 10.14569/IJACSA.2025.0161266.

[30] M. N. Alatawi, "Detection of fraud in IoT based credit card collected dataset using machine learning," Machine Learning with Applications, vol. 19, p. 100603, Mar. 2025, doi: 10.1016/j.mlwa.2024.100603.

[31] L. Abd-Ellatif, M. Abrar, and A. A. K. Ismaeel, "ATAD-Net: An Adaptive Deep Learning Framework for Real-Time Financial Fraud Detection," Advances in Artificial Intelligence and Machine Learning, vol. 05, no. 02, pp. 3988–4003, 2025, doi: 10.54364/AAIML.2025.52225.

[32] N. Thair Ali, S. Jasim Hasan, A. Ghandour, and Z. Salam Al-Hchimy, "Improving credit card fraud detection using machine learning and GAN technology," BIO Web Conf., vol. 97, p. 00076, Apr. 2024, doi: 10.1051/bioconf/20249700076.

[33] W. Y. Leong, Y. Z. Leong, and W. S. Leong, "Artificial Intelligence-Driven Fraud Detection: Enhancing Security in Digital Age," 2026, pp. 65–78. doi: 10.1007/978-981-95-2113-5_7.

[34] R. El Youbi, F. Messaoudi, M. Loukili, and R. Loukili, "From Click to Checkout: Deep Learning for Real-Time Fraud Detection in E-Payment

Systems," Statistics, Optimization & Information Computing, vol. 14, no. 6, pp. 3398–3408, Sep. 2025, doi: 10.19139/soic-2310-5070-2891.

[35] H. M. R. Al Lawati et al., "An Integrated Preprocessing and Drift Detection Approach With Adaptive Windowing for Fraud Detection in Payment Systems," IEEE Access, vol. 13, pp. 92036–92056, 2025, doi: 10.1109/ACCESS.2025.3569609.

[36] O. Pavlova, V. Askerov, B. Tomchyshen, and T. Hovorushchenko, "Intelligent AML systems: using deep learning to detect financial fraud," TalTech Journal of European Studies, vol. 15, no. 3, pp. 102–122, Dec. 2025, doi: 10.2478/bjes-2025-0032.

[37] S. Iqbal, K. M. Awan, S. Kamal, and Z. U. Rehman, "Interpretable Ensemble Learning Models for Credit Card Fraud Detection," Applied Sciences, vol. 15, no. 22, p. 12073, Nov. 2025, doi: 10.3390/app152212073.

[38] R. R. Devi, J. E. Raja, and Y. B. Chin, "Reinforcement learning with graph neural network (RL-GNN) fusion for real-time financial fraud detection: a context-aware community mining approach," Sci. Rep., vol. 15, no. 1, p. 42953, Dec. 2025, doi: 10.1038/s41598-025-25200-3.

[39] J. Černevičienė and A. Kabašinskas, "Explainable artificial intelligence (XAI) in finance: a systematic literature review," Artif. Intell. Rev., vol. 57, no. 8, p. 216, Jul. 2024, doi: 10.1007/s10462-024-10854-8.

[40] B. S. Prashanth et al., "Prediction of bank transaction fraud using TabNet—an adaptive deep learning architecture," International Review of Economics & Finance, vol. 106, p. 104916, Mar. 2026, doi: 10.1016/j.iref.2026.104916.

[41] C. O. Kungu, K. Senagi, and E. Omondi, "Hybrid deep learning for anti-money laundering: Unsupervised detection of emerging schemes via feature fusion and explainable artificial intelligence," Machine Learning with Applications, vol. 23, p. 100856, Mar. 2026, doi: 10.1016/j.mlwa.2026.100856.

[42] E. Karnavou, G. Cascavilla, G. Marcelino, and Z. Geradts, "I know you're a fraud: Uncovering illicit activity in a Greek bank transactions with unsupervised learning," Expert Syst. Appl., vol. 288, p. 128148, Sep. 2025, doi: 10.1016/j.eswa.2025.128148.

[43] R. K. Gupta et al., "Enhanced framework for credit card fraud detection using robust feature selection and a stacking ensemble model approach," Results in Engineering, vol. 26, p. 105084, Jun. 2025, doi: 10.1016/j.rineng.2025.105084.

[44] M. Hasan, M. S. Rahman, M. J. Morshed Chowdhury, and I. H. Sarker, "CNN Based Deep Learning Modeling with Explainability Analysis for Detecting Fraudulent Blockchain Transactions," Cyber Security and Applications, vol. 3, p. 100101, Dec. 2025, doi: 10.1016/j.csa.2025.100101.

[45] Y. Chen, C. Zhao, Y. Xu, C. Nie, and Y. Zhang, "Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications," Data Science and Management, Aug. 2025, doi: 10.1016/j.dsm.2025.08.002.

[46] G. Murugaraj, R. Krishnan, L. Harikrishnan, and M. Sellapan Kandasamy, "Systematic Review on the Influence of Classical and Quantum Machine Learning in the Financial Sector," Data Science and Management, Jan. 2026, doi: 10.1016/j.dsm.2026.01.001.

[47] Md. S. Mia, S. Roy, M. A. Ihsan, S. Hossain, and Md. K. U. Ahamed, "Data-driven financial fraud detection using hybrid artificial and quantum intelligence," BenchCouncil Transactions on Benchmarks, Standards and Evaluations, vol. 5, no. 4, p. 100252, Dec. 2025, doi: 10.1016/j.tbench.2025.100252.

[48] N. Uddin, "Role of AI in preventing financial crime: A comprehensive analytical review," Journal of Economic Criminology, vol. 13, p. 100200, Sep. 2026, doi: 10.1016/j.jeconc.2025.100200.

[49] A. Usman, N. Naveed, and S. Munawar, "Intelligent Anti-Money Laundering Fraud Control Using Graph-Based Machine Learning Model for the Financial Domain," Journal of Cases on Information Technology, vol. 25, no. 1, pp. 1–20, Jan. 2023, doi: 10.4018/JCIT.316665.

[50] E. A. L. Marazqah Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran, "A systematic review of literature on credit card cyber fraud detection using machine and deep learning," PeerJ Comput. Sci., vol. 9, p. e1278, Apr. 2023, doi: 10.7717/peerj-cs.1278.

[51] H. Ahmad, E. Rawashdeh, A. AlTawil, and N. Al-Ramah, "EFC-Tomek: An effective undersampling technique for credit card fraud detection," International Journal of Data and Network Science, vol. 9, no. 4, pp. 845–852, 2025, doi: 10.5267/j.ijdns.2025.7.003.

[52] Z. Miao, "Financial Fraud Detection and Prevention," Journal of Organizational and End User Computing, vol. 36, no. 1, pp. 1–27, Sep. 2024, doi: 10.4018/JOEUC.354411.

[53] A. A. Khan et al., "A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment," Computers and Electrical Engineering, vol. 102, p. 108234, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108234.

[54] Y. Wang, Q. Zheng, X. Li, L. Wang, and L. Lin, "CoSemiGNN: Blockchain fraud detection with dynamic graph neural networks based on co-association of semi-supervised," Expert Syst. Appl., vol. 298, p. 129853, Mar. 2026, doi: 10.1016/j.eswa.2025.129853.

[55] B. Lebichot, W. Siblini, G. M. Paldino, Y.-A. Le Borgne, F. Oblé, and G. Bontempi, "Assessment of catastrophic forgetting in continual credit card fraud detection," Expert Syst. Appl., vol. 249, p. 123445, Sep. 2024, doi: 10.1016/j.eswa.2024.123445.

[56] V. Chang, L. M. T. Doan, A. Di Stefano, Z. Sun, and G. Fortino, "Digital payment fraud detection methods in digital ages and Industry 4.0," Computers and Electrical Engineering, vol. 100, p. 107734, May 2022, doi: 10.1016/j.compeleceng.2022.107734.

[57] C. S. Kolli, V. Jagannathan, D. R. Lanka, C. Rajyalakshmi, V. K. R. Radha, and V. V. K. Reddy, "SCWO-SNN: Snow carpet weaver optimization based spiking neural network for fraud detection in UPI transactions using federated learning," Comput. Stand. Interfaces, vol. 98, p. 104166, Aug. 2026, doi: 10.1016/j.csi.2026.104166.

[58] V. Backiyalakshmi and B. Umadevi, "MRV-RSA: Developed Modified Random Value Reptile Search Algorithm and Deep Learning based Fraud Detection Model in Banking Sector," Data Knowl. Eng., vol. 163, p. 102557, May 2026, doi: 10.1016/j.datak.2026.102557.

[59] J. Diaz et al., "Dynamic Trust Modulation and Human Oversight in AI-Driven AML Systems: A Conceptual Framework for Compliance," International Journal of Advanced Computer Science and Applications, vol. 16, no. 12, 2025, doi: 10.14569/IJACSA.2025.0161209.

[60] A. Hosawi and R. Stone, "Enhancing Trust in Human-AI Collaboration: A Conceptual Review of Operator-AI Teamwork," International Journal of Advanced Computer Science and Applications, vol. 16, no. 7, 2025, doi: 10.14569/IJACSA.2025.0160701.

[61] E. Mill, W. Garn, N. Ryman-Tubb, and C. Turner, "Opportunities in Real Time Fraud Detection: An Explainable Artificial Intelligence (XAI) Research Agenda," International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, 2023, doi: 10.14569/IJACSA.2023.01405121.

Fusion of Large Language Model and Knowledge Graph for Intelligent Power-System Q&A

Zijie Meng, Xinlei Cai*, Lizhou Jiang, Kai Dong, Bingzhao Zhu, Junhong Guo
Power Dispatching and Controlling Center of Guangdong Power Grid Company Limited,
Guangzhou, Guangdong 510000, China

***Abstract*—Intelligent question-answering (Q&A) models are needed for modern power systems that can provide accurate, reliable, and explainable answers to engineers and operators. Conventional Q&A systems cannot always understand the question semantically or combine domain knowledge, and stand-alone Large Language Models (LLMs) can generate hallucinated or non-grounded answers. To tackle these challenges, this research introduces a fusion model that combines the power of LLMs with a Power Knowledge Graph (PKG) for intelligent power-system question answering. The proposed approach integrates LLMs' ability to understand language with that of structured knowledge representation by PKGs. First, z-score normalization is used to normalize the operational parameters, such as voltage, current, temperature, and load. After that, Bidirectional Encoder Representations from Transformers (BERT) is used to obtain semantic relations and domain-specific elements from textual data related to the power system. The PKG stores fault, operational, and maintenance knowledge, and a knowledge-enhanced retrieval mechanism extracts relevant information and adds it to the LLM prompt to boost the accuracy of reasoning and diminish hallucination. The model is assessed with the following tasks: Fault Type Identification, Environmental and Operational Causation, and Maintenance and Recovery Decision. The proposed Python-based implementation achieves BERT Score F1 values of 71.5%, 69.2%, and 67.8%, Faithfulness Scores of 64.8%, 61.5%, and 59.7%, and Hallucination Rates of 18.6%, 21.4%, and 24.1%, respectively. The results validate the proposed model's efficiency in intelligent power-system decision support through semantic consistency, factual accuracy, and context relevance.**

Keywords*—*Large language model; intelligent question answering; knowledge graph reasoning; smart grid, power systems; knowledge management

I. INTRODUCTION

Power systems are now highly interconnected and intelligent infrastructures enabling large-scale electricity generation, transmission, distribution, and consumption [1]. The complexity of power systems operations has risen dramatically due to the addition of renewable energy resources, distributed generation, smart meters, and automated control technologies [2]. Therefore, it is essential for power engineers and system operators to have efficient knowledge services that can deliver accurate information to ensure system operation, equipment maintenance, and fault diagnosis [3]. Intelligent question-answering (Q&A) systems have been identified as a key tool for enhancing knowledge accessibility and aiding in decision-making in power systems [4].

*Corresponding author.

Smart grids are being scaled up, and a wide variety of power equipment (generators, transformers, substations, transmission lines, and protection devices) is recording large amounts of operational and maintenance data [5]. The efficient use of this knowledge is critical for ensuring system reliability, maintaining continuous operation, and streamlining the process of fault recovery [6]. Traditional power-system Q&A approaches are mostly based on rules for retrieval and/or keyword-matching, but these have a known lack of semantic understanding, poor contextual reasoning, and limited adaptability to complex technical questions [7]. The constraints limit their ability to provide clear and understandable answers in dynamic operational contexts.

With recent advancements in LLMs, performance on natural language understanding and reasoning in context has proven promising for intelligent question-answering applications [8]. However, conventional LLMs may generate hallucinated responses, lack specific domain knowledge, have poor explainability, and are thus unsuitable for critical applications in power systems. Fortunately, a PKG can effectively represent the knowledge associated with power systems, which includes information on equipment, operating procedures, maintenance history, and fault occurrence [9]. Power System Faults Dataset. The proposed methodology enhances answer correctness, consistency, and explainability of the Q&A system by combining PKG with LLMs.

With the development of intelligent power system fault diagnosis and Q&A, knowledge graph-based models have been increasingly applied to improve structured knowledge retrieval and reasoning. An intelligent model for fault retrieval based on knowledge graphs was proposed by [10] for power communication equipment, with F1-scores of 81.7% and 87%, respectively, with multi-layer encoding and graph convolution. It demonstrated that it could effectively represent the semantics of a graph, but it had a heavy reliance on fault-specific data quality requirements. Table I presents a comparison of existing power-system fault analysis and decision-support methods.

Based on this motivation, this research suggests that LLM-PKG can be fused to realize intelligent power-system Question and Answer (Q&A). The model utilizes Knowledge-Enhanced Retrieval techniques to retrieve relevant entities, relationships, and operational records from the PKG and to incorporate them into the reasoning process of the LLM. Experimental assessment shows increased accuracy of answers, relevance to context, explanation, and answer consistency, and reduction of hallucination.

TABLE I. COMPARISON OF EXISTING METHODS FOR POWER-SYSTEM FAULT ANALYSIS AND KNOWLEDGE-BASED DECISION SUPPORT

Ref	Main Objective	Method	Results	Limitation
[11]	Fault analysis in Industrial IoT	Multi-Entity Knowledge Joint Extraction (MEKJE) with jointly trained models	F1-score: 78.6%	Limited generalization
[12]	Knowledge utilization for recommendation systems	Cross-domain knowledge graph embedding	Improved prediction accuracy	Scalability and computational complexity
[13]	Automatic electric-power fault knowledge extraction and Q&A	BERT–BiLSTM–CRF + electric power fault knowledge graph	Improved Chinese entity extraction and Q&A	Limited multilingual adaptability
[14]	Equipment fault-diagnosis named entity recognition	RoBERTa–BiLSTM–CRF (RBC)	F1-score: 94.98%	Limited dataset
[15]	Power-system fault detection and reasoning	Random Forest + knowledge reasoning	Accuracy: 90%	Small-scale IEEE 14-bus simulation limits real-world applicability
[16]	Power knowledge retrieval and inference	RPKR-GNN with GNN, user-intent fusion, and knowledge inference	Improved power-knowledge retrieval and reasoning	Limited explainability and scalability
[17]	Reliable LLM-based decision support for smart-grid operations	Locally hosted LLMs	Local LLMs outperformed standard LLMs in identifying unauthorized operations.	LLM hallucination and insufficient domain-specific knowledge grounding
[18]	Improve retrieval-augmented generation (RAG) accuracy and retrieval efficiency for complex queries.	Query complexity classification, keyword retrieval, semantic rules, and natural language inference (NLI)-based relevance checking.	On HotpotQA, accuracy improved by 9.4% and F1 by 16.14% over Adaptive-RAG.	Depends on keyword rules, custom training data, and NLI quality.
[19]	Improve retrieval and generation for complex multi-hop queries.	Domain-specific Knowledge Graph (KG) + Large Language Model (LLM) using entity extraction, path finding, query decomposition, semantic path ranking, and context aggregation.	Outperformed RAG and GraphRAG, especially for 5–6-hop queries, with strong directness.	Limited generalizability across domains.
[20]	Improve multi-hop reasoning over knowledge graphs.	Question-Directed Reasoning with Relation-Aware Graph Attention Network (QRGAT) using relation-aware GAT and reasoning graphs.	Improved Hits@1 by 2.3% on WebQuestionsSP and 3.6% on ComplexWebQuestions over NSM+h.	Computationally complex for multi-layer graph reasoning.
[21]	Improve online power-grid fault diagnosis and decision support.	Time-series density maps + Residual Network (ResNet) for equipment and fault classification.	Accurately identifies faulty equipment and fault type for online diagnosis.	Mainly designed for power-grid fault data; generalization may be limited.

Based on these investigations, this research observes that knowledge graphs provide structured knowledge representation and reasoning, whereas LLMs provide strong semantic understanding and natural-language generation. However, without combination, the reliable, interpretable, and consistent knowledge base still fails. The knowledge extraction approach in examination [16] uses the powerful ability of Graph neural networks to retrieve the most related facts from large corpora, the knowledge retrieval method also fails due to problems with explanation and scalability. Investigation [17] presents an LLM-based system, the smart grid knowledge management approach, but because LLMs can hallucinate, which brings down trust and reliability, and no domain knowledge is grounded in the models. Therefore, combining LLM and PKG is a potential solution for creating accurate, explainable, and intelligent power system Q&A systems. From an electrical engineering point of view, the proposed framework should support structured power-system knowledge and provide intelligent decision support. In contrast to existing question-answering systems based on languages, the proposed approach captures information about electrical operating conditions, equipment state, fault events, environmental conditions, and maintenance actions in the form of a structured PKG. These electrical relationships are then applied as proof in the framework for fault interpretation and the generation of maintenance-related responses. Thus, the contribution can be seen not only as LLM-based Q&A but as an intelligent information and decision-support model for power system fault analysis.

The proposed LLM–PKG combines structured power knowledge from the domain, while avoiding reliance on text retrieval or graph querying. Conventional RAG only retrieves the most relevant textual passages, while the proposed scheme retrieves structured triples representing equipment, operating variables, environmental conditions, faults, equipment states, and maintenance actions in the power system. Retrieval is based on BERT for power-domain entity identification and a one-hop neighborhood expansion performed in the power domain, followed by ranking candidates based on semantic similarity and graph-based electrical relationship preservation. The retrieval process differs from general GraphRAG methods by first using BERT to identify entities in the power domain, then applying the one-hop neighborhood expansion in the power domain, and subsequently ranking the candidate triples by semantic similarity while also retaining the graph-based electrical relationship information. The PKG is not meant to be the only system supporting a Q&A component of a knowledge graph; it is meant to be used as a fact-based system to provide evidence for an LLM that provides a natural language response in context. To this end, the contribution involved a task-specific combination of power-domain entity/relation extraction, structured retrieval of PKGs, relevance training of the search results provided to the LLM, and knowledge-guided LLM language generation of fault and cause analysis, as well as of maintenance/recovery options.

II. Power-System Problem Formulation

A. Electrical Operating Variables

Electrical, environmental, and equipment-condition variables such as voltage, current, power load, temperature, wind speed, fault duration, downtime, component health, and maintenance status determine the power-system operating state. Voltage and current define an electrical operating condition, and power load defines the level of loading of the power equipment or network. The temperature indicates thermal stress and overheating situations, while wind speed and weather conditions indicate environmental disturbance which may lead to equipment problems. Equipment condition and recovery information include component health, fault duration, downtime, and maintenance status. The proposed PKG and knowledge-grounded LLM for fault diagnosis and decision support rely on these variables as the electrical evidence.

B. Electrical Fault-Causality Representation

Faults in power systems are expressed in terms of causality relationships between electrical operating conditions, environmental conditions, equipment situations, events resulting in a potential fault, and maintenance needs. For instance, if the power load is increased, this can lead to higher current and thermal stresses; this might increase the temperature and reduce the material performance. Adverse weather, like windstorms or thunderstorms, can also add to electrical equipment and so increase the chances of fault occurrence. These dependencies are represented as triples of entity–relation–entity in the proposed PKG, which enables the LLM to rely on electrical evidence to deduce the relation, in addition to language-based evidence.

C. Electrical Fault-State Representation

The operating condition of the power system, along with the relevant electrical, environmental, and component-health parameters, is mapped in a fault-state vector to characterize the condition systematically and reliably for fault identification, as presented in Eq. (1):

$$X = [V,\ I, P, T, W, H, D_f,\ D_o] \quad (1)$$

where, V denotes the voltage, I is the current, P is the power load, T is the temperature, W is the wind speed/weather condition, H is the component health, D_f is the fault duration, and D_o is the downtime.

The electrical fault state is represented in Eq. (2) as a function of the observed operating state and structured power-system knowledge:

$$F = f(X, G) \quad (2)$$

where, F denotes the identified fault state, and G represents the PKG. For operator decision support, the generated response is formulated by Eq. (3):

$$Y = g(Q, X, G) \quad (3)$$

where, Q is the user query, and Y is the knowledge-grounded engineering response.

III. Methodology

The proposed methodology consists of sequential phases for developing an intelligent power-system Q&A model by integrating an LLM with a PKG. The model utilizes the Power System Faults Dataset as the foundational knowledge source and extends it through structured knowledge representation and retrieval-enhanced reasoning. As illustrated in Fig. 1, the Overall Architecture of the proposed LLM–PKG model for Intelligent Power-System Q&A.

A. Data Collection

The proposed model is based on the Power System Faults Dataset (https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset) that is obtained from Kaggle, which is used as a main knowledge source for intelligent power-system Q&A. There are 506 fault records, including operational, environmental, maintenance, and recovery data, such as voltage, current, power load, temperature, wind speed, weather conditions, component health, fault duration, and downtime. 80% of the records are used for training and 20% for testing. To ensure that the data partition is reproducible, the value 42, which is the same for all runs, is used as a fixed random seed. The evaluation looks at three Q&A tasks: Fault Type Identification, Environmental and Operational Causation, and Maintenance and Recovery Decision. Table II presents the categories of the dataset for structured fault reasoning: Fault Type Identification, Environmental and Operational Causation, and Maintenance and Recovery Decision. Sample Q&A pairs are presented in Table III for the purpose of replicably providing a benchmark of performance for the provided fault detection and decision support to be further compared and studied.

TABLE II. Power System Fault Dataset Category Overview

Category	Key Features Used	Fault Focus
Fault Type Identification	Voltage, Current, Power Load, Component Health	Line Breakage, Transformer Failure, Overheating
Environmental & Operational Causation	Temperature, Wind Speed, Weather Condition, Fault Duration	Weather-induced faults and operational triggers
Maintenance & Recovery Decision	Maintenance Status, Component Health, Downtime, Duration	Recovery urgency and maintenance prioritization

TABLE III. Representative Q&A Pairs from the Power Fault Dataset

Category	Question	Answer
Fault Diagnosis	What fault occurred in record F002?	The recorded fault was Transformer Failure, with faulty component health and 5 hours of downtime.
Environmental and Operational Analysis	What factors contributed to the overheating fault in F003?	The fault was associated with high temperature (35°C) and windstorm conditions, leading to component overheating.
Maintenance and Recovery Decision	What maintenance action is required for a transformer failure under faulty health status?	Immediate inspection and corrective maintenance are required to restore system stability and reduce downtime.

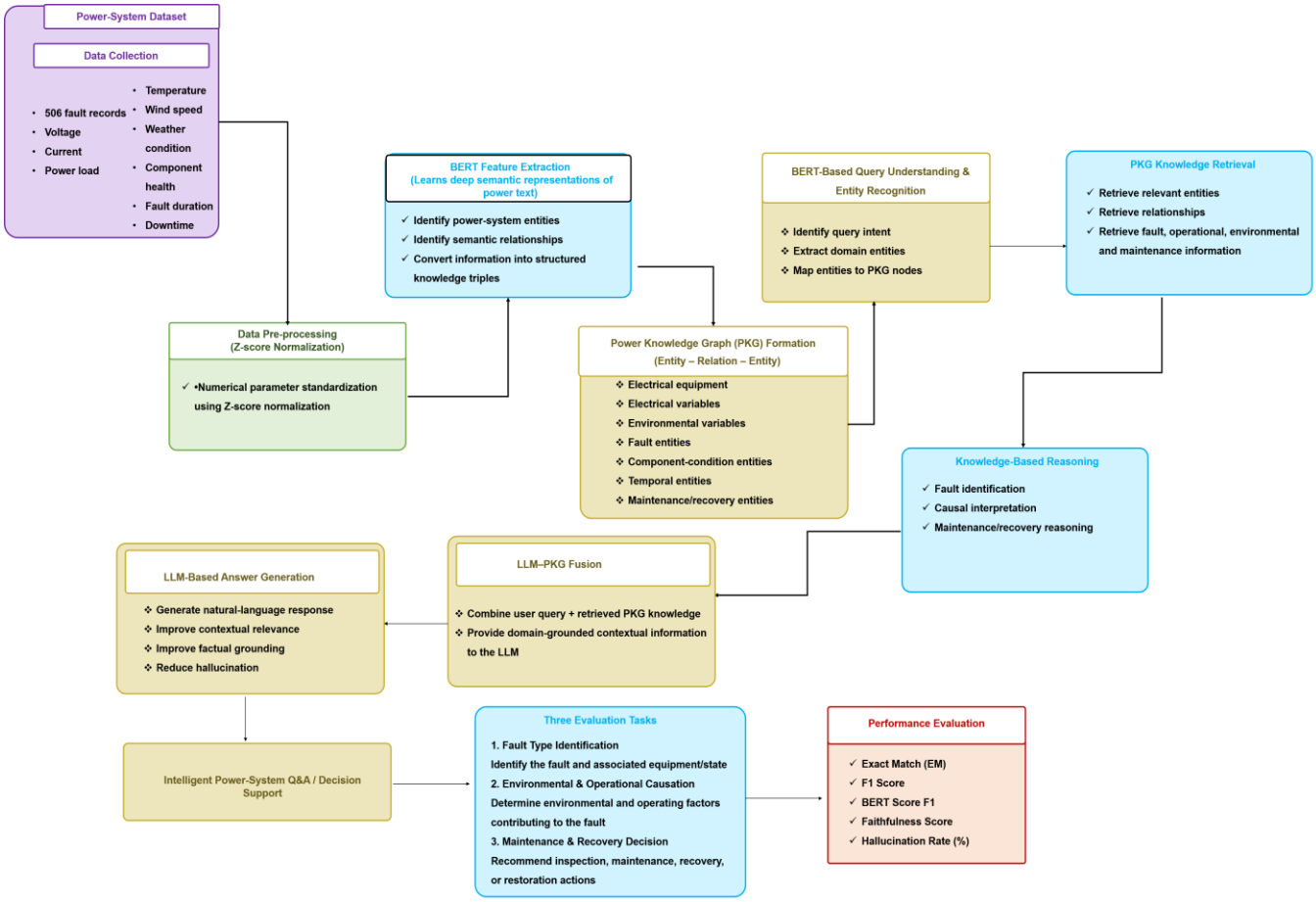


Fig. 1. Overall architecture of the proposed LLM–PKG fusion model for intelligent power-system Q&A.

The question–answer dataset is created from structured information in the 506 power-system fault records. For each of the following categories of semantic reasoning, questions were designed: fault diagnosis, electrical-state analysis, environmental and operational causation, maintenance and recovery, fault-duration and operational assessment, and LLM-based decision support. Each cluster of reasoning has the appropriate number of pairs of questions: 87, 85, 84, 83, 83, and 84 pairs of questions for the six thinking clusters. Each question has a fault record and structured PKG evidence associated. Answers to the ground truth questions are produced directly from the factual data attributes and relationships from the source records, such as downtime, maintenance status, fault type, equipment/component condition, electrical variables, environmental variables, fault duration, answers from the recovery requirements, and answers from the source records. The questions created are evaluated against the source record to make sure there is enough supporting information for each question to derive a deterministic source answer. Before assessing the question, the question is excluded from the examination if it has been duplicated or is unclear or not presented with the source record. The final reference answers are thus based on the original power-system records and associated PKG facts. These questions are coupled together in a question-answer set, which is subsequently applied in evaluating the accuracy of lexical matching, semantic similarity, factual grounding, and hallucination.

B. Data Pre-processing

This technique comes in handy for preprocessing and making heterogeneous numerical representations uniform before entity and relation extraction. Operating variables like voltage, current, power load, temperature, wind velocity, fault length, and time outage could range and have significantly varying numerical scales between different types of sensors. While these parameters can be utilized in the subsequent knowledge extraction process, their local scales can be different, so to scale down influence differences, they should be normalized using z-scores to convert them to a common scale, which allows for comparison of the same systems and easier representation of entities and relations in the power system. The transformation is defined in Eq. (4), where a normalized fault parameter is calculated based on its original value, the mean, and the standard deviation of the fault records. This normalization process helps maintain consistency among data and offers a stable numeric foundation for building the proposed PKG:

$$X = \frac{Z-\mu}{\sigma} \tag{4}$$

where, X is the normalized fault parameter value, and Z is the original operational parameter value, μ is the mean value of the same fault parameter, and σ is the standard deviation of the same fault parameter for all fault records. This normalization helps to maintain data consistency and to analyze and extract

knowledge from the faults that may occur within the proposed LLM-PKG system.

C. Power Knowledge Graph Construction

The PKG contains different interacting elements and relationships among the elements of electrical equipment, operating conditions, environment, fault events, and maintenance actions. The main entities are electrical equipment, fault types, electrical parameters, environmental conditions, component health types, maintenance types, and recovery conditions as shown in Table IV. These entities have causal, operational, spatial, temporal, and maintenance dependencies, known as relations. Power-System LLM–PKG Data Fusion Architecture integrating multi-modal operational data, Knowledge Graphs, and LLM-based reasoning is depicted in Fig. 2.

TABLE IV. ENTITY CATEGORIES AND REPRESENTATIVE ENTITIES IN THE POWER KNOWLEDGE GRAPH

Entity category	Representative entities
Electrical equipment	Transformer, transmission line, substation, relay, generator
Electrical variables	Voltage, current, power load
Environmental variables	Temperature, wind speed, weather
Fault entities	Line breakage, transformer failure, overheating
Condition entities	Component health, maintenance status
Temporal entities	Fault duration, downtime
Decision entities	Inspection, corrective maintenance, recovery

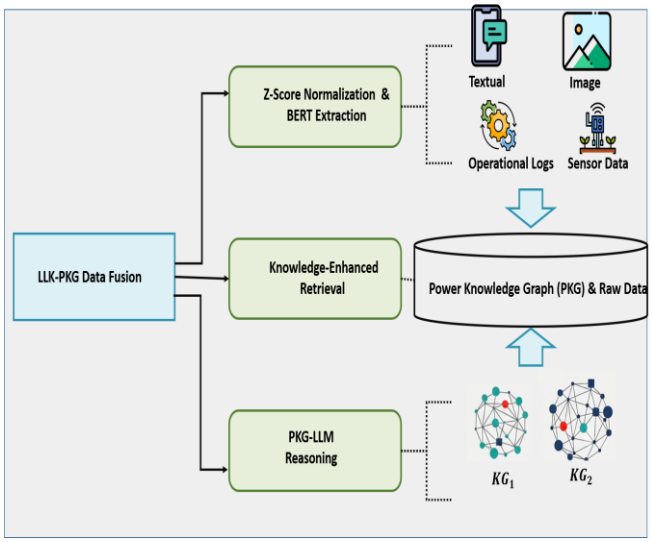


Fig. 2. Power-system LLM–PKG data fusion architecture.

The PKG is built based on the power-system fault records that have been normalized and deduplicated. The 506 normalized and deduplicated power-system fault records are used to build the PKG. It produces 1,248 distinct nodes, 1,836 entity-relationship-entity triples, 1,836 specific entity relationships, and 24 specific relation types. The types of nodes are electrical equipment, electrical variable, environmental, fault type, component, temporal, or maintenance or recovery decision. Relations express causal, operational, spatial, temporal, and maintenance dependencies. BERT will encode user query as a list of domain entities relevant to that query, mapping domain entities to the corresponding PKG nodes. The neighborhood system is then executed in a one-hop fashion to create potential candidate triples that are related to the extracted nodes. To rank candidate triples, the query representation is compared with representations of the triples, based on cosine similarity; thereafter, any duplicate triples are filtered out, and the top 5 triples are provided to the LLM as instructive and factual context. The process is dynamically linked to the query understanding process, the graph retrieval process, and the process of generating answers from the knowledge base.

The extracted entities are sorted based on their functional roles in the domain of power systems. Electrical equipment entities are physical objects like transformers, lines, substations, relays, and generators, while electrical and environmental variables are measurable operating conditions. Fault, condition, temporal, or decision entities offer more information on abnormal equipment health, fault duration, downtime, inspections, maintenance, and recovery. Finally, those entity pairs are translated into subject–relation–object triples and added to the PKG. This structured representation allows for enabling the connection of semantically relevant but different textual information by providing explicit semantic relations instead of handling only textual language. The resulting PKG can then be used as a knowledge layer when mapping queries to the graph and for evidence retrieval in the following step.

D. BERT-Based Power Entity and Relation Extraction

BERT is employed for extracting power-system entities and their semantic relationships from textual power-domain datasets. BERT is selected due to its bidirectional contextual learning capability, which enables effective understanding of technical terminology, operational dependencies, and fault-related semantics in power-system documents. The BERT input representation is composed of three embedding layers:

1) Token embedding: It transforms power-system textual tokens into dense vector representations. These embeddings capture domain-specific terms such as transformers, relays, substations, and fault types.

2) Input: The input to the BERT model consists of a linear sequence of tokens derived from the pre-processed power-system corpus. Special tokens [CLS] and [SEP] are added to mark the beginning and separation of input sequences.

3) Segment embedding: It distinguishes multiple sentence segments, which is useful for identifying inter-sentence relationships in fault reports and maintenance logs.

4) Position embedding: It preserves the sequential order of words in power-system text, enabling the model to understand the structural dependencies among entities and actions.

Using contextual embeddings, BERT extracts power-system entities and relations, converting fault, location, operational, environmental, maintenance, and temporal information into structured knowledge triples for PKG.

The semantic extraction module extracts power-system entities and relations from textual descriptions of the power system and user queries using BERT-base-uncased. The contextual BERT representation is then input to a token-classification output layer to perform entity classification. After the identification of entities, entity pairs are candidates in the considered textual context and then categorized based on the power-system relation inventory defined. The extracted entity–relational information is coded as structured triples (head entity, relation, tail entity) and added to the PKG. The same BERT representation process is used during the query processing stage for domain entity identification and to aid in their mapping to the respective nodes in the PKG. The final answer generation is then left to BERT. It is designed to deliver meaning to natural-language queries and to link them to the structured knowledge of the power system. The LLM then gets the evidence retrieved by the PKG and produces the completed natural-language result.

E. User Query Processing and Entity Recognition

In this phase, the proposed model receives natural language queries related to power-system faults, operational conditions, and maintenance decisions. The query is processed using BERT-based semantic analysis to understand user intent, capture contextual information, and identify domain-specific entities and relationships. The extracted entities are then aligned with the corresponding nodes in the PKG, enabling accurate knowledge retrieval and supporting subsequent reasoning and response generation.

F. Contextual Query Models

To illustrate the query understanding process, representative contextual Q&A examples derived from the power fault dataset are presented in Fig. 3 and Fig. 4. These examples demonstrate how fault records are transformed into structured contextual inputs for semantic interpretation. The contextual representation enables BERT to capture fault-specific dependencies, operational conditions, environmental influences, and maintenance-related information from user queries. This process supports effective entity recognition and improves the alignment between user queries and Power Knowledge Graph entities for accurate knowledge retrieval and reasoning.

Context 1: Fault record F002 reports a Transformer Failure at coordinates *(34.056, -118.245)* under rainy weather conditions. The transformer operated at 1800 V, 180 A, and 45 MW load with a recorded temperature of 28°C. The component health status was marked as Faulty, with a fault duration of 3 hours and a total downtime of 5 hours. Maintenance history indicates the component was previously serviced.
Question: Which operational parameter indicates that the transformer failure was likely associated with equipment degradation rather than thermal overload?
Answer: The moderate operating temperature (28°C) combined with the Faulty health status suggests the failure was caused by internal equipment degradation rather than excessive thermal stress.
Question: Why does the downtime exceed the fault duration in F002?
Answer: The longer downtime (5 hrs) compared to the fault duration (3 hrs) indicates additional time was required for fault isolation, transformer inspection, and restoration of grid connectivity.

Fig. 3. Contextual query examples for power fault analysis for fault diagnosis and operational analysis.

Context 2: Fault F003 records an Overheating event at (34.0525, -118.244) during windstorm conditions. The system operated at 2100 V, 230 A, and 55 MW load. The temperature reached 35°C, and the component health was classified as Overheated. The fault persisted for 4 hours, resulting in 6 hours of downtime.
Question: Which combination of parameters most strongly indicates thermal stress as the primary cause of failure?
Answer: The elevated temperature (35°C), high power load (55 MW), and Overheated component condition collectively indicate thermal stress as the dominant fault-driving factor.
Question: How could windstorm conditions contribute to overheating despite not being a direct thermal source?
Answer: Windstorm conditions can introduce mechanical stress and cooling instability, reducing heat dissipation efficiency and increasing the probability of sustained thermal accumulation in the component.

Fig. 4. Contextual query examples for power fault analysis for environmental and thermal fault reasoning.

G. LLM-PKG for Intelligent Power-System Question Answering

The proposed LLM-PKG model integrates structured knowledge retrieval with language generation for intelligent power-system question answering. After query entity recognition, relevant knowledge triples are retrieved from the PKG and combined with the user query as contextual input to the LLM. This hybrid retrieval-reasoning mechanism improves answer accuracy, contextual relevance, and explainability, and reduces hallucinations through domain-grounded response generation. The proposed model employs Llama-3.1-8B-Instruct (8 billion parameters), a decoder-only Transformer LLM, for knowledge-grounded answer generation, while BERT-base-uncased is used for entity and relation extraction and query understanding. The LLM is used in pretrained form without additional domain-specific pretraining. For each query, BERT identifies relevant entities, which are mapped to PKG nodes. A two-stage graph-neighborhood and semantic retrieval method is then applied, using cosine similarity between query and triple embeddings. The top 5 relevant triples (K=5) are retrieved and inserted into a structured prompt containing the task instruction, user query, and retrieved PKG evidence. The PKG constructed from the 506 fault records contains 1,248 nodes, 1,836 edges, 24 relation types, and 1,836 unique triples. Since the LLM is used without fine-tuning, training epochs and learning rate are not applicable to the LLM. During inference, the temperature is set to 0.2, with a maximum generation length of 256 tokens. The implementation uses Python 3.11, PyTorch 2.3.1, and Transformers 4.44.2, and experiments are performed on an NVIDIA RTX 4090 (24 GB), Intel Core i9-13900K CPU, and 64 GB RAM. A fixed random seed of 42 is used to ensure reproducibility.

The fusion process includes four consecutive steps: Query interpretation, evidence retrieval in the graph database, contextual fusion, and response generation. The first is the BERT query processor, which detects user intents, entities, and relationships. Second, the identified entities are mapped onto corresponding nodes in PKG, and their graph neighborhoods are searched to get related knowledge triples. Third, recovered triples are reduced and structured into contextual evidence with the original query. Lastly, the query-evidence context is given to the LLM for answer generation, where the instruction is tiled.

Finally, the query-evidence context is presented to the LLM for answer generation, with an instruction tiled around it. This design decouples the structured knowledge retrieval from the natural-language generation and enables the LLM to rely on explicitly available evidence from the power system for response generation. Language understanding and generation in LLMs should therefore remain unchanged, with the aim of enhancing the factual grounding via the fusion mechanism.

1) Power Knowledge Graph (PKG) retrieval and reasoning: After query entity recognition, the extracted entities are mapped to their corresponding nodes in the PKG to retrieve relevant structured knowledge. The retrieval process explores entity–relation connections associated with fault events, operational parameters, environmental conditions, and maintenance records. This is followed by semantic reasoning on extracted knowledge triples to infer fault causes, operating dependencies, and maintenance needs. This process enhances context understanding, ensuring the generated response is based on knowledge in power systems. The constructed PKG contains 1,248 unique entities (nodes), 24 relation types, and 1,836 unique entity–relation–entity triples represented by 1,836 directed edges. The graph was constructed from the 506 power-system fault records after entity normalization and duplicate removal. The entities include electrical equipment, electrical variables, environmental conditions, fault types, component conditions, temporal information, and maintenance/recovery decisions. For each query, the identified entities are mapped to corresponding PKG nodes, and their connected relations and triples are retrieved to provide domain-specific evidence for LLM-based reasoning.

To ensure a repeatable knowledge-retrieval approach, the suggested LLM–PKG strategy involves the semantic retrieval phase based on a query, the neighborhood expansion in the retrieved graph, and the relevance-based triple selection. The recognized entities and the representation of the query are fed into the retrieval system after the query is processed using BERT. Both the textual description and the semantic embedding of each PKG entity and each PKG relation are used. Both the query embedding and the embeddings for candidate entity–relation–entity triples are compared with cosine similarity in Eq. (5):

$$S(q, t_i) = \frac{\mathbf{q} \cdot \mathbf{t}_i}{\|\mathbf{q}\| \|\mathbf{t}_i\|} \quad (5)$$

where, (q) denotes the user-query representation and $\mathbf{t}_i$ denotes the semantic representation of the $i - th$ knowledge triple. A higher cosine-similarity value indicates greater semantic relevance between the query and the corresponding PKG triple. There are two steps in the retrieval process. The entities in the query are first matched to the review nodes corresponding to the entities in PKG, and candidate triples are extracted that have a connection with these nodes by a 1-hop graph traversal. Second, the candidate triples are ordered by the cosine-similarity score. Only the top-K (K is set by the user) highest-ranked triples are stored in order to avoid irrelevant information being passed to the LLM. The proposed implementation assumes that K=5; that is, it supplies at most five knowledge triples to the LLM that are relevant to the same query. All triples are left in if fewer than 5 satisfy the relevance criterion. Semantic similarity is the main criterion for determining the final selection score, leaving the graph relationship criterion for power-system reasoning unchanged. So, the triple is chosen if it meets both conditions: 1) it is connected with at least one query-relevant entity or its graph neighborhood and 2) it meets a high enough similarity score compared to other candidate triples. Triples are deleted unless they represent duplicates, and semantically redundant triples are removed prior to the final top-(K) selection. This approach involves minimizing the contextual data provided to the LLM while keeping the electrical, environmental, fault, and maintenance connections pertinent to the query.

The retrieval process can be summarized as follows: 1) receive the natural-language query; 2) recognize domain entities and query intent with BERT; 3) match the recognized entities to the PKG nodes; 4) conduct a one-hop neighborhood expansion to get candidate triples; 5) produce semantic representations for the query and candidate triples; 6) calculate the similarity of the query with candidate triples based on cosine similarity; 7) rank the candidate triples by the cosine similarity score in descending order; 8) filter duplicate or redundant triples; 9) use the top five triples as the knowledge context input to the LLM to generate the answer. The retrieved triples are then appended to the original query to follow a prompt template as outlined in 3.7.2. This retrieval mechanism acts as a direct link between understanding the query, the graph-based retrieval of evidence, and reasoning with the LLM. For instance, if a query about transformer overheating is issued, the overload retrieval will return triples related to the transformer equipment along with temperature, overheating, component health, weather, and maintenance actions, but not unrelated power-system entities such as circuit breakers or an index. This results in the LLM only being given a subset of the evidence relevant to each particular query, thereby increasing the relevance of the material and minimizing the likelihood of unsupported generation.

Only the graph evidence most closely related to the semantic content of the query should be retained during the retrieval stage. Once entity recognition is complete, the retrieved anchors are nodes of the corresponding PKG. Their neighbouring triples give the evidence for the candidates in the form of descriptions of equipment states, fault conditions, environmental influences, temporal characteristics, and maintenance actions. Finally, the retrieved evidence is ranked by semantic similarity between the query representation and candidate triples. Top-ranked triples are kept as knowledge in context, because redundant triples are meant to be stripped away to reduce repetition of context and unnecessary details. This retrieval approach minimizes the information that the LLM receives in the graph and focuses the response on the most relevant facts within the world of the power system.

2) Large Language Model (LLM)-based answer generation: The retrieved knowledge triples after retrieval and inference are supplemented with the user query as input for generating answers by the LLM. The LLM is able to produce accurate, context-aware, and explainable responses based on the natural language query combined with the structured PKG

context. This knowledge-guided generation enhances semantic understanding and mitigates hallucination by embedding domain-specific evidence. The proposed model uses Llama-3.1-8B-Instruct, with 8 billion parameters, as the LLM for knowledge-grounded answer generation. The model follows a decoder-only Transformer architecture and is used in its pretrained, instruction-tuned form without additional fine-tuning on the power-system dataset. The 506 power-system fault records are used to construct the PKG and generate the knowledge-grounded Q&A context, rather than being used for additional LLM pretraining. BERT is used separately for power-system entity/relation extraction and query understanding. The top 5 relevant PKG triples are retrieved using semantic similarity and appended to the query as contextual evidence. The instruction requires the LLM to answer using the retrieved power-system facts and avoid unsupported information. Since the LLM is not further fine-tuned, training epochs, learning rate, and fine-tuning method are not applicable. During inference, the temperature is set to 0.2, the maximum generated length is 256 tokens, and deterministic decoding is used to improve reproducibility. Thus, the LLM functions as the final knowledge-grounded response generator, while the PKG supplies the domain-specific factual evidence.

To evaluate the generated answers, the 20% test set is compared to the corresponding reference answers in the Power System Faults Dataset. Exact Match (EM) indicates verbatim accuracy of the generated answer and the reference. F1 is used to assess the accuracy of generated answers on a token-by-token basis. BERT Score F1 is the version of BERT that compares answers based on contextual representations. Faithfulness measures how consistent the responses are with the retrieved PKG evidence, while Hallucination Rate measures the percentage of responses that include information that is unsupported or incorrect. All comparison variants use the same testing subset, reference answers, retrieval settings, and evaluation criteria as are utilized for the proposed LLM–PKG model. The scores obtained are then averaged across the assessed test set on each of the three Q&A scenarios, quantifying the correctness of the answers, semantic similarity, factual grounding, and answering behavior of hallucinated answers.

Llama-3.1-8B-Instruct (8B parameters), pretrained in an instruction-tuned fashion, has been used for the implementation without fine-tuning. BERT-base-uncased is used alone for entity and relation extraction, and for query understanding. Power-System does not, therefore, update the parameters of the LLM, but rather its records are mapped to structured, evidence-of-expertise power-system instances, as well as question–answer instances associated with knowledge. Given a query, BERT finds entities that are relevant to the query, aligns the entities by mapping them to the nodes in PKG, and expands the one-hop neighborhood for formulating candidate triples. Corresponding to the query representation, a group of top 5 (K=5) triples is returned, with the highest-ranking one according to cosine similarity between the query and the corresponding triple representations. The LLM is provided with a structured prompt that consists of the following three parts: 1) an instruction that asks the LLM to respond in a relevant way that can be supported with evidence; 2) the original user query; and 3) the retrieved PKG triples. The answer must be limited to the information provided in the Power System Supplies and does not include statements of facts that are not supported by the information provided. To evaluate a model for inference, they set the temperature to 0.2 and the maximum generation length to 256, and set the decoding method to deterministic to be reproducible. No further fine-tuning, optimization, or further training epochs are applied, both on the LLM and during the preprocessing. It is implemented on an NVIDIA RTX 4090 (24 GB memory), an Intel Core i9-13900k (CPU), and 64 GB of RAM with Python 3.11, PyTorch 2.3.1, and Transformers 4.44.2. In the experimental implementation, a random seed of 42 is fixed throughout the implementation. The cross-entropy training loss is not applied to the LLM in the proposed experimental setting, as is the norm with the LLM in isolation. The generation stage is thus handled as inference, where a pre-trained instruction-tuned model is conditioned on the retrieved PKG evidence. The cross-entropy loss function is used to optimize the LLM during training, measuring the difference between the probabilities of the tokens generated by the LLM and the actual target sequence as shown in Eq. (6):

$$K = -\sum_{i=1}^{m} \log P\left(x_j | y, x_1, x_2, \dots, x_{j-1}\right) \qquad (6)$$

where, K denotes the cross-entropy loss; m is the total answer length, y represents the knowledge-augmented input (user query + PKG facts), x_j is the jth generated token, log denotes the logarithmic function measuring prediction confidence, and $\log P\left(x_j | y, x_1, x_2, \dots, x_{j-1}\right)$ is the conditional probability of generating the next answer token.

H. Electrical Fault Diagnosis and Decision-Support

This retrieved knowledge in turn is leveraged to make three engineering decisions: fault identification, causal interpretation, and maintenance/recovery recommendation. The query is initially converted into electrical bodies and working parameters. The relevant PKGs are then retrieved to look for supporting evidence. This LLM then uses the retrieved electrical evidence and the user question to formulate the final response. The response obtained is then compared with the fault information and the decision requirements. The proposed model thus couples machine measurements and equipment conditions to structured fault knowledge and natural-language decision support.

IV. Results

The experimental evaluation aimed to determine the language-based reasoning capability and assess the electrical fault-diagnosis capability of the proposed LLM–PKG model. This evaluation focuses on fault identification; causal factors associated with the environment and operation; and decision-making on maintenance and recovery. Moreover, measures of semantic similarity and knowledge-grounding, the performance in electrical fault diagnosis is assessed with the application of classification-oriented measures. The Power System Faults Dataset is then statistically analysed and validated to assess its structural quality and appropriateness for intelligent power-system question answering. The pairs of Q&A were then analyzed in a rule-based system using Python, where the results

were grouped into three categories: Fault Type Identification, which includes fault detection, category, and location; Environmental and Operational Causation, which classifies environmental and operational factors that cause faults; and Maintenance and Recovery Decision, which identifies corrective actions, maintenance planning, and restoration of systems. The distribution of the dataset by category, shown in Table V and Fig. 5, is a well-balanced distribution that supports the different aspects of the fault reasoning and decision-making process.

TABLE V. DISTRIBUTION OF QUESTION CATEGORIES ACROSS THE POWER SYSTEM FAULTS DATASET

Question Category	Fault Type Identification (%)	Environmental & Operational Causation (%)	Maintenance & Recovery Decision (%)
Diagnostic Reasoning	38.2	12.4	8.6
Causal Analysis	14.5	41.3	9.2
Operational Assessment	21.8	28.6	14.1
Maintenance Planning	6.4	4.8	39.5
Recovery Evaluation	5.1	3.9	21.4
Decision Support	14.0	9.0	7.2

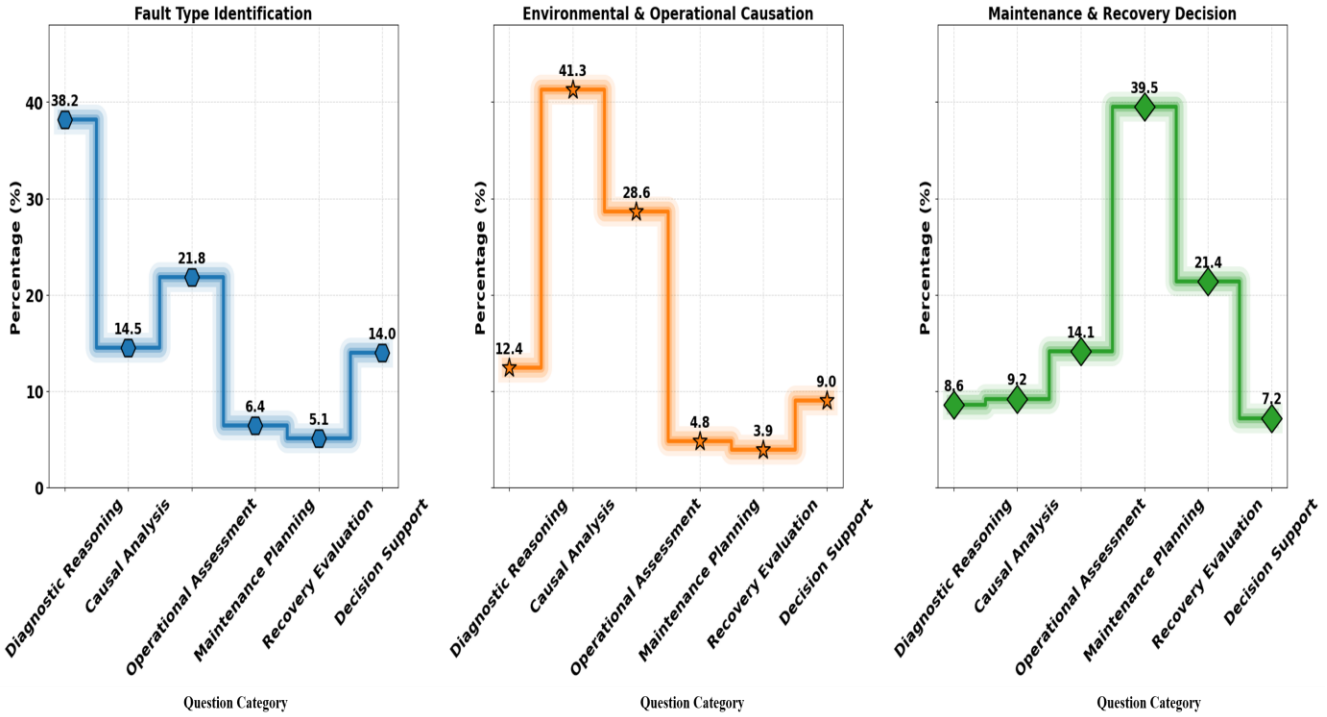


Fig. 5. Category-wise distribution of power-system fault reasoning tasks.

The dataset exhibits a well-balanced distribution across six defined reasoning categories. The questions generated from the texts were clustered into six clusters of semantic reasoning, based on the dominant information content and reasoning requirements of queries. There are a total of 87, 85, 84, 83, 83, and 84 pairs of Q&A generated in the six clusters, respectively, for a total of 506 pairs across all six clusters. Overall cluster proportions varied from 16.27% to 17.26%, suggesting a reasonably even balance of types of questions within the clusters that focused on fault diagnosis, electrical-state analysis, environmental and operational causes, maintenance and recovery, duration of fault, and operational decision support using LLM.

Table VI summarizes the properties of the dataset, and Fig. 6 outlines the Semantic Network of the principal entities related to power system faults. Power system fault diagnosis, electrical system state analysis, power system condition evaluation, power system maintenance management, power system fault loss assessment, and fault analysis on decisions supported by LLM are the six main categories of questions on which the Power System Faults Dataset displays an even distribution. Such a distribution of question types allows rich reasoning abilities for fault diagnosis, maintenance planning, fault repair, system restoration, and operation decisions in the power system context. The generated question set was grouped into six semantic reasoning clusters according to the dominant information and reasoning requirements of the queries. The cluster sizes were relatively comparable, indicating a balanced distribution of question types.

TABLE VI. DISTRIBUTION OF CLUSTER ID ACROSS THE POWER SYSTEM FAULTS DATASET

Cluster ID	Size (Fault Records)	Top Defining Terms	Sample Query
Cluster 1	87	fault type, transformer failure, line breakage, overheating, fault location, component health	What type of fault occurred in F002, where was it located, and what caused it?
Cluster 2	85	voltage, current, power load, electrical parameters, electrical state	How did voltage, current, and power load conditions contribute to the fault event in F001?
Cluster 3	84	weather condition, rainy, snowy, thunderstorm, windstorm, environmental factors	Which weather conditions contributed to transformer overheating in F003?
Cluster 4	83	maintenance status, scheduled, pending, completed, maintenance actions, recovery process	What is the maintenance status of the affected component, and what actions are required after the fault?
Cluster 5	83	fault duration, downtime, outage duration, system availability, recovery time	How long did the fault last, and what was the resulting downtime of the system?
Cluster 6	84	LLM-based intelligent Q&A, fault diagnosis answer, fault interpretation, risk interpretation, operational decision support	What operational decisions are recommended based on the diagnosed fault and associated risks?
Total	506	—	—

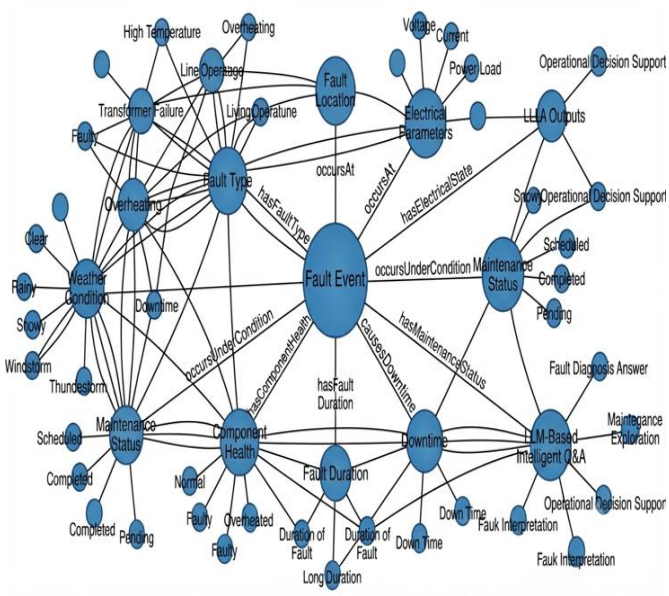


Fig. 6. PKG representation of thematic fault clusters and entity relationships.

A. Performance Evaluation

The performance of the novel proposed fusion model LLM-PKG is assessed using three functional areas: Fault type detection; Causation and origin environment and operations; and support and decision making for operation and maintenance. Both lexical and semantic analysis of output metrics, and measures for Knowledge grounding and for hallucination evaluation.

1) Exact Match (EM): Measures whether the generated answer exactly matches the reference answer after normalization. It is defined as shown in Eq. (7):

$$EM = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\,(\hat{y}_i = y_i) \tag{7}$$

where, N is the number of evaluated questions, $\hat{y}_i$ is the generated answer, y_i is the ground-truth answer, and $\mathbb{I}(\cdot)$ is an indicator function. It assigns a value of one only when the prediction completely matches the reference.

2) Token-level F1-score: Evaluates partial lexical overlap between generated and reference answers. Precision (P), recall (R), and F1 are calculated as given in Eq. (8)-(10):

$$P = \frac{|T_{\hat{y}} \cap T_y|}{|T_{\hat{y}}|} \tag{8}$$

$$R = \frac{|T_{\hat{y}} \cap T_y|}{|T_y|} \tag{9}$$

$$F1 = \frac{2PR}{P+R} \tag{10}$$

where, $T_{\hat{y}}$ and T_y represent the predicted and reference token sets, giving partial credit when the response contains only part of the expected information.

3) BERTScore: Measures contextual semantic similarity using the RoBERTa-large pretrained checkpoint as the contextual encoder. Pairwise cosine similarities between candidate token x_i and reference token y_j are calculated. The final BERTScore is reported as Eq. (11):

$$F1_{BS} = \frac{2P_{BS}R_{BS}}{P_{BS}+R_{BS}} \tag{11}$$

This accommodates technically correct power-system answers that use alternative wording while preserving semantic meaning. $2P_{BS}$ specifies the BERTScore precision. R_{BS} denotes BERTScore recall. $F1_{BS}$ represents the harmonic mean of precision and recall, indicating the overall contextual semantic similarity between the predicted and reference answers.

4) PKG-Based Faithfulness Score (FS): Serves as the primary measure of factual grounding for the LLM–PKG architecture. Generated answers are decomposed into atomic claims ($C = \{c_1, c_2, \ldots, c_m\}$) and compared against retrieved PKG facts ($K = \{k_1, k_2, \ldots, k_r\}$). Claim-level faithfulness is calculated using Eq. (12).

$$FS = \frac{N_{supported}}{N_{claims}} \times 100 \tag{12}$$

where, $N_{supported}$ is the number of supported claims and N_{claims} is the total number of claims. High scores indicate statements regarding fault types, equipment conditions,

environmental factors, or maintenance actions are directly supported by PKG evidence.

5) Hallucination Rate (HR): Quantifies unsupported content generated by the LLM shown in Eq. (13):

$$HR = \frac{U}{A} \times 100 \quad (13)$$

where, U is the number of unsupported factual answer units and A is the total number of factual answer units. A lower rate indicates better grounding.

The Faithfulness Score (FS) represents the ratio of factual claims, stated in the answer generated by the model, that are backed up by the retrieved evidence/items from the PKG. Each answer is broken down into individual pieces of factual information (fingers) such as: fault type, equipment condition, environmental factor, operational condition, or recommended maintenance action. A claim is faithful if there exists evidence in the retrieved PKG triples or the related source record to the claim. The Faithfulness Score is therefore equal to the ratio of the number of claims that get supported and the total number of factual claims.

The Hallucination Rate is the percentage of factual statements that cannot be substantiated with the evidence from the PKG retrieved, or the corresponding source record. There is a possibility that one or more variables of the unsupported claim may be incorrect: the fault classified as the fault type which is not the true cause; the operating condition which is not the true cause; an environmental cause not observed in the evidence; a maintenance recommendation which is not supported by the available power-system data. Therefore, a lower HR means a more solid basis of facts.

In addition to automated grounding measures, domain-expert assessment is used because numerical and semantic text measures are not sufficient to determine if the response is technically correct with respect to decision-making in power systems. Each answer needs to be evaluated by a specialist for its factual accuracy, relevance to the specific electrical circumstance for which the inquiry was made, consistency among the retrieved evidence, and engineering suitability of the suggested actions. The expert assessment considers and reports on linguistic similarity without also reporting on technical content and validity of the domain-specific content, which are reported separately from it. The results of all three tasks are shown in Table VII.

TABLE VII. DISTRIBUTION OF QUESTION CATEGORIES ACROSS LLM-PKG-BASED POWER SYSTEM FAULT DATASET

Metric	Fault Type Identification	Environmental & Operational Causation	Maintenance & Recovery Decision
EM	62.4%	58.7%	55.9%
F1 Score	68.1%	65.3%	63.4%
BERT Score F1	71.5%	69.2%	67.8%
Faithfulness Score	64.8%	61.5%	59.7%
Hallucination Rate	18.6%	21.4%	24.1%

The proposed model gets the best score in Fault Type Identification because fault types have a structured and clear pattern; its performance decreases slightly in Environmental and Operational Causation due to a complex system because of the dependency on various factors (e.g., weather, load). The scores in Maintenance and Recovery Decision tasks are lower as the need for high-order deduction across various system characteristics increases; the consistently good results in BERTScore F1 scores (ranging from 67.8% to 71.5%) confirm a high semantic similarity to ground truth; Faithfulness scores in these tasks confirm the excellent effect of PKG fact grounding. The hallucination rate (ranging from 18.6% to 24.1%) shows a relatively low rate thanks to knowledge-graph fusion. To assess the contribution of the main elements of the suggested LLM–PKG fusion framework, an ablation study was carried out. The whole model was contrasted with variations that did not include PKG retrieval and reasoning, BERT-based semantic extraction, Z-score normalization, or an LLM-only baseline. The findings in Table VIII show that eliminating these elements increases hallucination while decreasing answer accuracy, fidelity, and semantic similarity. Specifically, a greater performance loss results from the elimination of PKG retrieval and reasoning, underscoring the need for domain-grounded knowledge for trustworthy power-system Q&A and decision support.

TABLE VIII. ABLATION ANALYSIS OF LLM–PKG COMPONENTS FOR POWER-SYSTEM Q&A

Model / Variant	Z-score	BERT Extraction	PKG Retrieval & Reasoning	LLM Fusion	EM (%)	F1 (%)	BERTScore F1 (%)	Faithfulness (%)	Hallucination (%)
Rule-based Q&A	✗	✗	✗	✗	43.5	49.1	52.8	55.6	34.2
PKG-only retrieval	✗	✗	✓	✗	46.8	52.4	56.7	57.2	31.6
LLM-only baseline	✗	✗	✗	✗	51.2	57.6	61.5	53.8	28.1
Conventional RAG	✓	✓	✗	✓	56.7	62.5	66.1	59.3	23.7
Without PKG retrieval/reasoning	✓	✓	✗	✓	54.6	60.8	64.3	56.9	25.3
Without BERT extraction	✓	✗	✓	✓	58.9	64.2	67.8	60.7	21.8
Without Z-score normalization	✗	✓	✓	✓	59.8	65.4	69.1	61.9	20.7
Full LLM–PKG	✓	✓	✓	✓	62.4	68.1	71.5	64.8	18.6

The advancement of intelligent power system fault diagnosis and Q&A systems leverages KG-based reasoning as a means to enrich structure information retrieval and enhance explainability. Numerous works prove that KG-augmented

models achieve high accuracy in the task of fault retrieval and entity-relation extraction from the domain-specific query based on KG, but struggle with generalization capability, dependence on labeled data quality, and scalability issues in real-world large, complex power grid scenarios [14]. KG embedding and KG-reasoning models can further obtain satisfactory results regarding prediction performance; however, this comes with computational expense and lower efficiency in massive knowledge contexts [13].

Smart grid operation with LLMs makes considerable improvements in reasoning efficiency but is deficient in hallucination and shallow-grounded structured domain knowledge to guarantee safe decision-making. For this reason, the LLM-PKG fusion model is designed by fusing context-language modeling with structured knowledge graphs that enhance the factual grounding of predictions and their inferential logic correctness. Extensive experiments verify the model's robust effectiveness across all benchmarks that reflect semantic understanding, strengthen knowledge grounding, boost response faithfulness, and mitigate hallucination. The obtained results validate and prove the effectiveness of the proposed framework, LLM-PKG, in providing truthful and trustworthy answers to intelligent question answering in a power system with context. The model performs well in fault type identification with fixed pattern and relatively poor performance in complex logical inference with dependent variables as multiple factors, respectively. Several limitations also existed in terms of limitations of data scale, extraction noise in the construction of PKG, and insufficient temporal inference abilities. Future directions involve real-time and scalable applications, graph learning, and multi-modality. The proposed model is designed to be a tool for the decision-making process of power-system engineers and operators. It offers a natural language user interface to which one can pose questions on conditions for the electrical operation, on faults, on the environment, and on maintenance. The PKG now structures electrical evidence, and the LLM is used to interpret and generate the evidence to be formulated into a response. The model thus does not stand alone as a protection or control tool, but is designed to be advisory in nature.

V. Conclusion

With the gradual development of the power system, the research and application of intelligent Q&A models have become more and more important in the field of power system fault diagnosis and effective operation decision support. Traditional methods are difficult to reason with and lack structured reasoning ability, and standalone LLMs have limitations in factually grounded reasoning and hallucination. To overcome these limitations, the proposed LLM–PKG fusion model combines z-score normalization, BERT-based entity and relation extraction, and Power Knowledge Graph-guided retrieval to boost knowledge-driven reasoning. The experimental results for all tasks (Fault Type Identification, Environmental and Operational Causation, and Maintenance and Recovery Decision) are 71.5%, 69.2%, and 67.8%, respectively, with the Hallucination Rate of <25% for all of them (18.6%, 21.4%, and 24.1%). The results demonstrate that structured graph knowledge, alongside LLM reasoning, has a substantial impact on the semantic consistency, factual correctness, and explainability of questions and answers within a power system. The electrical fault-analysis experiments illustrate how the model can link operating measurements to environmental conditions, structure, and equipment states, and incorporate fault knowledge and natural-language decision support. The model would serve as a practical basis for intelligent fault diagnosis and smart grid decision support, and future research involves real-time data integration, temporal reasoning, and the scaling of graph learning for widespread use.

Acknowledgment

Supported by the Science and Technology Projects of China Southern Power Grid Corporation, Project Number : 036000KC24080003 (GDKJXM20240652).

References

[1] Abdelkader S, Amissah J, Abdel-Rahim O. Virtual power plants: an in-depth analysis of their advancements and importance as crucial players in modern power systems. Energy Sustain Soc 2024;14(1):52. https://doi.org/10.1186/s13705-024-00483-y

[2] Zeb N, Zulkifli SA, Sepeeh MS, et al. Generalized Unified Power Flow Controller for Low Oscillation Damping for Smart Grid Using Adaptive Centralized Control. J Electr Eng Technol 2026;21(3):2207–2222. https://doi.org/10.1007/s42835-026-02590-6

[3] Rosati R, Romeo L, Cecchini G, et al. From knowledge-based to big data analytic model: a novel IoT and machine learning-based decision support system for predictive maintenance in Industry 4.0. J Intell Manuf 2023;34(1):107–121. https://doi.org/10.1007/s10845-022-01960-x

[4] Gao M, Li M, Ji T, et al. Key technologies of intelligent question-answering system for power system rules and regulations based on improved BERTserini algorithm. Processes 2023;12(1):58. https://doi.org/10.3390/pr12010058

[5] Gupta R, Chaturvedi KT. Adaptive energy management of big data analytics in smart grids. Energies 2023;16(16):6016. https://doi.org/10.3390/en16166016

[6] Yang S. Transformer model optimization method for multi-modal data fusion. Int J Adv Comput Sci Appl 2025;16(7). https://doi.org/10.14569/IJACSA.2025.0160708

[7] Zhang P, Li J, Chen S, et al. Multi-feature fusion for relation extraction using entity types and word dependencies. Int J Adv Comput Sci Appl 2023;14(7). https://doi.org/10.14569/IJACSA.2023.0140731

[8] Kosasih EE, Margaroli F, Gelli S, et al. Towards knowledge graph reasoning for supply chain risk management using graph neural networks. Int J Prod Res 2024;62(15):5596–5612. https://doi.org/10.1080/00207543.2022.2100841

[9] Kim J, Podlasek A, Shidara K, et al. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. Sci Rep 2025;15(1):39426. https://doi.org/10.1038/s41598-025-22940-0

[10] Qu H, Zhang Y, Liang K, et al. A knowledge-graph-driven method for intelligent decision-making on power communication equipment faults. Electronics 2023;12(18):3939. https://doi.org/10.3390/electronics12183939

[11] Liang K, Zhou B, Zhang Y, et al. A multi-entity knowledge joint extraction method of communication equipment faults for Industrial IoT. Electronics 2022;11(7):979. https://doi.org/10.3390/electronics11070979

[12] Liu J, Huang W, Li T, et al. Cross-domain knowledge graph chiasmal embedding for multi-domain item-item recommendation. IEEE Trans Knowl Data Eng 2022;35(5):4621–4633. https://doi.org/10.1109/TKDE.2022.3151986

[13] Meng F, Yang S, Wang J, et al. Creating a knowledge graph of electric power equipment faults based on a BERT–BiLSTM–CRF model. J Electr Eng Technol 2022;17(4):2507–2516. https://doi.org/10.1007/s42835-022-01032-3

[14] Gao F, Zhang L, Wang W, et al. Named entity recognition for equipment fault diagnosis based on RoBERTa-wwm-ext and deep learning

integration. Electronics 2024;13(19):3935. https://doi.org/10.3390/electronics13193935

[15] Li C, Wang B. A knowledge graph method towards power system fault diagnosis and classification. Electronics 2023;12(23):4808. https://doi.org/10.3390/electronics12234808

[16] Hou R, Zhang Y, Ou Q, et al. Recommendation method of power knowledge retrieval based on graph neural network. Electronics 2023;12(18):3922. https://doi.org/10.3390/electronics12183922

[17] Imanifard A, Majidi B, Shamisa A. SmartGridAgent: an educational framework for reliable digital twin-based smart grid workforce training with locally hosted LLMs. Smart Grids Sustain Energy 2025;10(2):41. https://doi.org/10.1007/s40866-025-00274-0

[18] Huang J, Wang M, Cui Y, et al. Layered query retrieval: An adaptive framework for retrieval-augmented generation in complex question answering for large language models. Applied Sciences 2024;14(23):11014. https://doi.org/10.3390/app142311014

[19] Osipjan A, Khorashadizadeh H, Kessel AL, et al. Graphtrace: A modular retrieval framework combining knowledge graphs and large language models for multi-hop question answering. Computers 2025;14(9):382. https://doi.org/10.3390/computers14090382

[20] Zhang G, Liu J, Zhou G, et al. Question-directed reasoning with relation-aware graph attention network for complex question answering over knowledge graph. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2024;32:1915–1927. https://doi.org/10.1109/TASLP.2024.3375631

[21] Zhang X, Ding R, Wang Z, et al. Power grid fault diagnosis model based on the time series density distribution of warning information. International Journal of Electrical Power & Energy Systems 2023;146:108774. https://doi.org/10.1016/j.ijepes.2022.108774

Adapting Ephemeral ECC Authentication for Decentralized Web 3.0: A Threat Model and Security Analysis

Adarsh S V Nair*, Rathnakar Achary
Alliance School of Advanced Computing, Alliance University, Bengaluru, India

***Abstract*—Web 3.0 shifts control of digital identity and data away from centralized platforms, moving authentication onto resource-constrained end-user devices, wallets, and smart contracts. RSA and ElGamal are computationally expensive at the key sizes required for adequate security, making them a poor match for this setting. This study presents a user-centric Web 3.0 authentication framework combining Elliptic Curve Cryptography with an ephemeral Elliptic Curve ElGamal (ECEG) character-encoding scheme, building on an ephemeral-encoding design for constrained IoT devices and its subsequent hash/XOR authentication extension. Unlike that prior design, the mutual-authentication step here binds each party to its long-term key through digital signatures and replaces static-key ECEG transport with a two-ephemeral-key ECEG transport, providing an erasure-conditional forward-secrecy property that the prior design lacks. We define a Dolev-Yao threat model with an in-scope/out-of-scope adversary boundary, an assumption-based security analysis, a formal-verification specification, and a decentralized-identifier/verifiable-credential mechanism binding long-term public keys to claimed identities, addressing the identity-binding gap that otherwise limits man-in-the-middle resistance without a common certificate authority. A protocol-level operation count and a complexity comparison against RSA are reported without claiming empirical superiority. We situate the framework against its closest prior IoT-oriented work and active 2025–2026 literature on ECC- and blockchain-based decentralized authentication. Recent 2026 analyses have reduced estimates of the quantum resources needed to solve the elliptic-curve discrete logarithm problem underlying secp256k1-style curves; we give a post-quantum outlook against current NIST migration guidance. The evaluation remains analytical rather than empirical; formal verification and hardware benchmarking remain necessary before production deployment.**

Keywords*—Elliptic curve cryptography; decentralized authentication; Web 3.0 security; privacy-preserving authentication; self-sovereign identity; decentralized identifiers; verifiable credentials; forward secrecy; formal verification; resource-constrained devices; post-quantum cryptography

I. Introduction

The development of the World Wide Web is generally described in three overlapping phases. Web 1.0 was a static medium in which users were essentially passive consumers of information provided by publishers. Web 2.0 swung the pendulum towards participation, allowing for social interaction and user-generated content, but then conversely centralized data, identity, and economic value in the hands of a few large platforms. Web 3.0 is best understood as a correction to this concentration: a computing paradigm that uses blockchain infrastructure, cryptographic identity primitives, and peer-to-peer protocols to return control of data and digital identity to the people who created it.

This change of control does not remove the need for security; it moves it. In Web 2.0, a central identity provider can enforce authentication policy, patch vulnerabilities, and revoke compromised credentials for its users. In a decentralized environment, this burden falls on end-user devices, wallets, and smart contracts, which usually have limited computational power, memory, and energy budgets. Decentralized identity technologies such as Decentralized Identifiers (DIDs), Self-Sovereign Identity (SSI), and Zero-Knowledge Proofs (ZKPs) give individuals cryptographic control of credentials without a centralized issuing authority. These mechanisms are attractive because they minimize third-party trust, but they also raise the bar for the cryptographic building blocks that underpin them: authentication must be robust enough to withstand active adversaries, yet lightweight enough to run on constrained hardware such as embedded wallets, IoT gateways, and mobile edge devices participating in decentralized applications.

This tension has practical consequences that are visible in the attack landscape that has grown alongside Web 3.0. Social-engineering attacks such as rug pulls, Sybil attacks, and phishing trick users into giving up credentials or authorizing malicious transactions, and weak or spoofable authentication is a direct contributor to several of these: if the link between a user and their cryptographic identity can be spoofed, replayed, or intercepted, an attacker can impersonate a legitimate party or enrol duplicate identities under someone else's claimed identity. Smart contracts are self-executing and largely immutable once deployed, which makes them susceptible to logic errors and reentrancy attacks, and cryptojacking silently hijacks computing power for unauthorized mining; these two threats are primarily software-security and resource-abuse problems rather than authentication failures, and the framework proposed here does not address them (Section X). Strengthening the authentication layer is therefore not a complete defense for Web 3.0 but a necessary precondition for the identity-dependent parts of its security promises [1].

Public-key cryptosystems such as RSA and ElGamal have been the standard building blocks for mutual authentication between distributed entities for decades, but their computational profile, dominated by large-modulus modular exponentiation,

*Corresponding author.

and their reliance on large keys make them poorly suited to resource-constrained devices operating within decentralized networks. Elliptic Curve Cryptography (ECC) is an alternative that provides comparable security with a much smaller key size, resulting in less computation time, a smaller memory footprint, and lower communication overhead. These properties make ECC a natural candidate for authentication in Web 3.0 settings, provided that the protocol built around it is designed with the specific trust assumptions of decentralized environments in mind.

This study presents a user-centric authentication framework for Web 3.0 that uses Elliptic Curve Cryptography together with an ephemeral Elliptic Curve ElGamal (ECEG) character-encoding scheme, a two-ephemeral-key ECEG parameter transport, and a digital-signature-based mutual authentication step. The ephemeral ECEG character encoding builds on the encoding concept originally proposed for constrained IoT devices by Baccouri et al. [2], and the goal of binding the resulting encoding parameters to an authenticated exchange builds on that group's own subsequent extension of the idea, which transports the encoding parameters inside an ECEG ciphertext under the recipient's static public key and confirms them with a hash-and-XOR exchange chained to the previous session's parameters, verified with the Scyther tool and benchmarked against RSA [3]. As Section II-D discusses in detail, this study departs from [3] at the protocol level as well as the application level: mutual authentication here rests on digital signatures over each party's long-term key rather than on the hash-and-XOR confirmation of [3], and the encoding parameters are transported under a fresh, per-session ephemeral key rather than under the recipient's static public key. Under the stated erasure assumption, this removes the static-key forward-secrecy limitation of [3] (Section VI-D). The present study's contribution is therefore both this protocol-level adaptation and the Web 3.0-specific threat model, requirements analysis, and identity-binding mechanism built around it, for the setting in which two entities may share no prior relationship and no common certificate authority. The contributions of this work are summarized as follows.

- A review of the authentication and identity-management challenges specific to decentralized Web 3.0 architectures, linking them to concrete attack patterns reported in the literature.
- An explicit, requirement-driven threat model for Web 3.0 peer-to-peer authentication, together with an honest positioning of this work relative to its closest prior art (Section II-D).
- A Web 3.0-oriented adaptation of an existing ECEG character-encoding scheme [2,3], replacing its static-key encoding transport with a two-ephemeral-key ECEG exchange and its mutual-authentication step with digital signatures over each party's long-term key, reanalyzed for the trust assumptions of self-sovereign, peer-to-peer Web 3.0 identity. The adaptation is designed to operate without exposing long-term private keys and provides an erasure-conditional forward-secrecy property that the prior static-key design lacks.
- An informal, assumption-based security analysis against eavesdropping, man-in-the-middle, replay, and impersonation attacks – with man-in-the-middle resistance stated explicitly as conditional on authenticated public-key binding rather than as an unqualified guarantee – together with a concrete formal-verification plan (Section VI-F) that specifies the exact queries needed to move from an informal argument to a machine-checked one.
- A concrete decentralized-identifier/verifiable-credential (DID/VC) mechanism (Section VI-F) for binding long-term public keys to claimed identities, closing the residual key-to-identity binding gap that otherwise limits man-in-the-middle resistance in a peer-to-peer setting with no common certificate authority.
- A computational complexity comparison between RSA and ECC, checked against the current NIST SP 800-57 equivalence table, together with an honest discussion of the limits of the present analytical-only evaluation and concrete directions for hardware-based validation.
- A post-quantum outlook (Section IX) that situates the proposed ECC-based framework against the rapidly narrowing 2025–2026 quantum resource estimates for solving the elliptic-curve discrete logarithm problem, and against current NIST migration guidance for algorithm deprecation.

The remainder of this study is organized as follows. Section II reviews related work on public-key and elliptic-curve-based authentication for constrained and decentralized systems, and positions this work relative to its closest prior art. Section III introduces the mathematical preliminaries of elliptic curve cryptography used throughout the study. Section IV defines the threat model and security objectives. Section V presents the proposed authentication framework in detail. Section VI gives the security analysis, including the formal-verification plan. Section VII studies computational complexity relative to RSA. Section VIII discusses the results. Section IX gives a post-quantum outlook. Section X discusses limitations and future work, and Section XI concludes the study.

II. Related Work

A. Public-Key Cryptography as a Baseline for Authentication

Traditionally, public-key cryptosystems such as RSA and ElGamal have been used to achieve mutual authentication between distributed entities, where the security of the scheme rests on the difficulty of factoring large integers or computing discrete logarithms in large multiplicative groups. These systems require key sizes of two to four thousand bits or more for an adequate security margin, and the modular exponentiation operations involved are computationally intensive. This overhead is manageable on servers and general-purpose computers, but becomes a bottleneck on the memory- and energy-constrained devices common in Internet of Things (IoT) deployments, and increasingly in Web 3.0 client software such as wallets and decentralized-application agents. This limitation motivates the search for cryptographic primitives that realize similar security with a smaller computational footprint.

B. Elliptic Curve Cryptography for Constrained and Decentralized Systems

Elliptic Curve Cryptography is increasingly the preferred choice for constrained environments because it offers security comparable to RSA with much shorter key lengths, owing to the higher per-bit strength of the Elliptic Curve Discrete Logarithm Problem (ECDLP). ECC has been used for authentication in resource-constrained settings in a large body of work. Dinarvand and Barati [4] proposed an efficient and secure ECC-based RFID authentication protocol; Lamrani Alaoui et al. [5] proposed a highly efficient ECC-based authentication protocol for RFID tags under similarly tight resource constraints. Noori et al. [6] extended this line of work to the healthcare Internet of Things with an ECC-based mutual authentication scheme for RFID-enabled medical devices, and Sowjanya et al. [7] designed an ECC-based authentication scheme specifically for the Internet of Medical Things. Khan et al. [8] proposed a secure framework combining authentication and encryption with an improved ECC construction for IoT-based medical sensor data, illustrating the recurring role of healthcare IoT as a proving ground for lightweight ECC protocols.

ECC authentication using smart cards and multi-factor schemes has also been heavily explored. Kumari et al. [9] proposed a smart-card-based mutual authentication protocol named ESEAP, and a related work by the same group [10] proposed a secure user authentication protocol based on ECC. Li et al. [11] proposed a robust and energy-efficient authentication protocol for the industrial Internet of Things, while Panja et al. [12] combined ECC with fingerprint biometrics to strengthen user binding. Ramasamy et al. [13] proposed an ECC-secured one-time-password scheme for mobile commerce; Sadri and Asaar [14] designed an anonymous two-factor authentication protocol for IoT applications with emphasis on user privacy; Elias et al. [15] implemented a one-time-password system on top of ECC key exchange; and Kaur et al. [16] proposed a two-factor authentication framework for cloud computing based on the same underlying elliptic-curve operations. A common observation across these contributions is that identity-based and certificate-based ECC schemes can fail either because they lack user anonymity, or because the authentication step itself remains vulnerable to replay and impersonation if ephemeral values are reused or predictable a gap this work directly addresses by using fresh ephemeral parameters on a per-session basis.

On the implementation side, Islam et al. [17] proposed high-performance ECC processors with unified point addition on twisted Edwards curves, and Basu Roy and Mukhopadhyay [18] presented a high-speed implementation of ECC scalar multiplication over prime fields for generic Montgomery curves, both demonstrating that the scalar multiplication at the heart of ECC can be realized efficiently even on constrained hardware. Lara-Nino et al. [19] provide a comprehensive survey of elliptic-curve lightweight cryptography that catalogs the design trade-offs between security margin, key size, and implementation cost that motivate the present work. The most directly related prior work is that of Baccouri et al. [2], who proposed an ephemeral encoding scheme in which plaintext characters are mapped onto curve points using randomly generated ephemeral parameters rather than a fixed, exchanged lookup table. The framework proposed in this study builds on that ephemeral-encoding idea; Section II-D discusses in detail how it relates to that group's own subsequent extension of the same idea into a full authentication protocol [3].

The field has remained active through 2025 and into 2026. Prakash and Ramesh [20] proposed a decentralized ECC-based authentication scheme with a dynamic difficulty-adjustment mechanism for IoT networks, correctness argued with BAN logic and evaluated against replay and man-in-the-middle attacks, architecturally the closest 2025 comparator to the framework proposed here. Liu et al. [21] introduced EdenDID, an edge-computing and blockchain-based decentralized identity system for Web3 applications that binds a human user, wallet address, and physical device together specifically to resist Sybil attacks on Web3 wallets, and Le et al. [22] combined blockchain smart contracts with AI-driven document recognition and Merkle-tree-based access control for decentralized identity management. These 2025 contributions confirm that ECC-based and blockchain-based decentralized authentication remains a fast-moving area, which strengthens the case, developed in Section II-D, for differentiating this study's contribution narrowly and explicitly rather than on the underlying cryptographic primitives alone.

C. Blockchain, Identity, and Verification

Beyond point-to-point authentication, several works consider the broader decentralized identity and trust ecosystem in which such protocols operate. Achary et al. [1] proposed an integrated blockchain model for personally identifiable information systems, examining how blockchain-based identity management can mitigate the social-engineering and identity-theft risks characteristic of Web 3.0 platforms, and a related study [23] investigated the integration of ECC with the Constrained Application Protocol (CoAP) to secure resource-constrained IoT communication, a combination architecturally close to the decentralized, low-power devices targeted here. A related blockchain-enabled IoT logistics system for high-value shipment tracking [52] similarly pairs a smart contract with blockchain-based access control, illustrating the same decentralized-trust pattern applied to a supply chain rather than an authentication setting, and a multi-authentication smart-locker design [53] combines several authentication modalities in a deployed IoT system; both are system-level rather than protocol-level contributions, but they reinforce the point that authentication should be evaluated alongside privacy mechanisms and deployment cost rather than through key-size comparisons alone. Zheng et al. [24] proposed a blockchain-based lightweight authentication and key-agreement scheme for the Internet of Vehicles, illustrating how ECC-style lightweight authentication is increasingly paired with distributed ledgers rather than centralized authentication servers. Formal verification of such protocols is itself an active area: Cremers [25] developed Scyther, a tool for the automated verification of security protocols, and Blanchet [26] developed ProVerif, based on the applied pi calculus; both tools are used to motivate the formal-verification plan of Section VI-F. Finally, K. K. G. V. et al. [27] explored machine-learning-based anomaly detection for cloud data, a complementary line of defense that could, in future work, be layered on top of the authentication mechanism

proposed here to flag statistically anomalous authentication attempts rather than only individually valid or invalid ones.

Within the same domain, Kanjanapruthipong and Boonkrong [28] proposed a blockchain-based decentralized authentication paradigm for closed environments, with security argued using Gong–Needham–Yahalom (GNY) logic rather than the Dolev–Yao-style symbolic verification pursued here, illustrating the range of formal approaches the field currently uses. Recent systematization work has also matured: a companion comparative survey [29] in this journal analyzes the scalability, security, and feasibility trade-offs between centralized and decentralized identity management, and a 2026 systematization of knowledge [30] organizes the open challenges of self-sovereign digital identities across a broad cross-section of the recent literature.

D. Positioning Relative to the Closest Prior Art

The protocol most closely related to the one proposed here is the lightweight authentication scheme of Baccouri et al. [3], which extends their own earlier ephemeral-encoding proposal [2] by exchanging the ephemeral encoding parameters (α, β) inside an ECEG ciphertext C encrypted under the recipient's static public key, sending C together with a hash value h1 in the first message and returning a hash-and-XOR confirmation CH1 = h3 ⊕ h1 whose hash input includes the previous session's encoding parameters; the resulting protocol is verified with the Scyther tool — checking secrecy, liveness, weak agreement, non-injective agreement and synchronization, and commitment to the exchanged values — and benchmarked against RSA on a C implementation across 8-, 64-, and 256-bit inputs. At the level of the cryptographic mechanism, that scheme and the framework proposed in this study therefore share the ephemeral ECEG character-encoding step of [2]; the mutual-authentication mechanism itself differs, since [3] authenticates via a hash-and-XOR confirmation that depends on state carried over from the previous session rather than via digital signatures over the current transcript, and transports the parameters under a static recipient key rather than the two-ephemeral-key ECEG transport used here (Section V-C). Table I sets these differences out message by message. Based on the openly available full text of [3], that protocol is designed and evaluated for classical IoT and wireless-sensor-network deployments between two devices provisioned in advance, and it does not address the specific trust assumptions of Web 3.0: there is no notion of a self-sovereign, wallet-held identity, no discussion of decentralized identifiers, and no attempt to reason about anonymity in the sense required by self-sovereign identity systems.

The contribution of the present study relative to [3] operates at two levels. At the protocol level, mutual authentication here rests on digital signatures over each party's long-term key combined with a two-ephemeral-key ECEG transport of the encoding parameters (Section V-C), replacing the static-key ECEG transport and previous-session-chained hash/XOR confirmation of [3], and providing an erasure-conditional forward-secrecy property that [3] does not have (Section VI-D). At the requirements level, the contribution is: 1) an explicit, requirement-driven threat model formulated for the trust assumptions of decentralized Web 3.0 identity, where the two parties may share no prior relationship and no common certificate authority (Section IV); 2) a concrete decentralized-identifier/verifiable-credential mechanism for binding long-term public keys to claimed identities (Section VI-D), together with an accompanying discussion of anonymity, pseudonymity, and forward secrecy framed in self-sovereign-identity terms rather than device-provisioning terms (Section VI-E); and 3) a concrete, ready-to-execute formal-verification plan with explicit secrecy and injective-agreement queries (Section VI-F), intended to be carried out as direct follow-up work. Readers should treat the ephemeral ECEG character-encoding step of Section V-B as a design pattern inherited from [2] and [3]; the signature- and two-ephemeral-key-ECEG-based mutual-authentication mechanism built around it (Section V-C), the Web 3.0-specific requirements analysis, the identity-binding mechanism, and the verification plan are this study's own contribution, summarized in Table I.

TABLE I. MESSAGE-LEVEL AND TRUST-MODEL DIFFERENCES BETWEEN THE CLOSEST PRIOR SCHEME [3] AND THE PROTOCOL OF SECTION V

Aspect	Baccouri et al. [3]	This study (Section V)
ECEG character encoding	Ephemeral (α,β) encoding of [2]	Retained unchanged; inherited, not claimed as new
Parameter transport	ECEG ciphertext C under recipient's static public key	ECEG ciphertext (C_1, C_2) under recipient's fresh per-session ephemeral point R_b
Message 1	C, h1	C, N_A, σ_A = Sign_{k_A}(hash(C ‖ N_A ‖ Q_A ‖ Q_B ‖ R_b))
Message 2	CH1 = h3 ⊕ h1 (hash input includes previous session's parameters)	N_B, σ_B = Sign_{k_B}(hash(M ‖ N_A ‖ N_B ‖ Q_A ‖ Q_B))
Authentication primitive	Hash and XOR	Schnorr/BIP-340 signatures over the current transcript
First-session bootstrap	Depends on previous-session state	Stateless per session; valid signature required regardless of history
Replay protection	Session chaining	Fresh 128-bit nonces + bounded replay cache
Forward secrecy	Not provided (static-key transport)	Provided conditionally if A erases i,r and B erases r_b after single use (Section VI-D)
Identity binding	Devices provisioned in advance	DID/VC binding check before the handshake (Section VI-B)
Added per-session cost	—	+1 ephemeral point, +2 signature ops, +2 verifications (Table VI); not claimed lighter than [3]
Validation	Scyther-verified; RSA benchmark	Analytical; verification plan (Section VI-F) and benchmarking pending

E. Research Gap

The literature makes two things clear: 1) ECC is well suited to authentication on constrained devices because of its favorable security-to-key-size ratio; and 2) ephemeral, non-repeating cryptographic parameters are important for resisting replay and impersonation. What is comparatively unexplored, as Table II summarizes, is the combination of these two properties analyzed against the specific trust requirements of Web 3.0 identity management, where authentication must also support user-

controlled, self-sovereign identity without a centralized issuing or verifying authority. Existing ECC-based authentication schemes, including the closest prior art discussed in Section II-D, are analyzed for classical IoT, RFID, or wireless-sensor-network environments rather than for the decentralized application and wallet interactions of Web 3.0. The framework proposed in this study targets that gap. This remains true of the most recent 2025–2026 literature identified during the preparation of this revision (Table II): the closest 2025 comparators [20,21,28] each address either dynamic difficulty adjustment, blockchain-level decentralized authentication, or Sybil-resistant wallet binding, but none combines ephemeral character-level ECEG encoding with a signature- and two-ephemeral-key-ECEG-based mutual-authentication handshake analyzed against a Web 3.0-specific threat model, which is the specific combination this study targets.

TABLE II. SUMMARY OF REPRESENTATIVE ECC-BASED AUTHENTICATION SCHEMES

Ref.	Target Domain	Core Mechanism	Noted Limitation
[4,5]	RFID	ECC-based mutual authentication for tags	Tight tag-side resource budget limits protocol complexity
[6–8]	Healthcare IoT / IoMT	ECC authentication and encryption for sensor data	Emphasis on confidentiality; limited anonymity guarantees
[9,10]	Smart-card authentication	ECC-based mutual authentication with smart cards	Card loss/cloning risk; anonymity depends on card security
[12]	Biometric authentication	ECC combined with fingerprint features	Requires biometric sensing hardware
[13,15]	OTP-based authentication	ECC key exchange for one-time-password generation	OTP channel itself can be a phishing target
[14]	IoT two-factor authentication	Anonymous two-factor protocol	Second factor adds device/hardware dependency
[2]	IoT messaging	Ephemeral ECC/ECEG character encoding (no authentication handshake)	No explicit mutual authentication handshake
[3]	IoT / wireless sensor networks	Ephemeral ECEG encoding + hash/XOR confirmation chained to previous session; Scyther-verified; RSA-benchmarked	Targets classical IoT/WSN; no Web 3.0/SSI trust model or DID/VC binding
[20]	IoT networks (decentralized)	ECC + dynamic difficulty adjustment; BAN-logic proof	No ephemeral character-encoding layer; not Web 3.0/SSI-specific
[28]	Closed blockchain environments	Blockchain-based decentralized authentication; GNY-logic proof	ECC used only for validator-node key agreement, not for end-user authentication
[21]	Web 3.0 wallets / DePIN	Edge computing + blockchain identity binding human, wallet, and device	Targets Sybil resistance specifically; no ephemeral encoding or signature-based parameter-transport handshake
This work	Web 3.0 / decentralized identity	Ephemeral ECEG encoding + signature/two-ephemeral-key ECEG mutual authentication and DID/VC binding, adapted to an explicit Web 3.0-oriented threat model	Forward secrecy added over [3]; evaluated analytically; formal verification and hardware benchmarking pending (Section X)

III. ELLIPTIC CURVE CRYPTOGRAPHY

Anonymity is an essential requirement for authentication in open networks, especially where the privacy of the interacting parties must be protected. Early designs for mutual authentication in public-key cryptosystems such as RSA and ElGamal relied on resource-intensive modular exponentiation and a separate infrastructure for certificate management. Fig. 1 illustrates how steeply this modular-exponentiation cost grows with key size under a naive-multiplication model. This combination is difficult to justify on devices with limited processing power, memory, or battery capacity, which motivates the use of Elliptic Curve Cryptography as a lighter-weight alternative with an equivalent security guarantee.

An elliptic curve over a finite field $\mathbb{F}p$ is formally the set of points (x, y) that satisfy $y^2 = x^3 + ax + b$, together with a distinguished point at infinity that serves as the identity element. The points on the curve form an Abelian group under a geometrically defined addition rule. In this group, scalar multiplication is repeated addition of a base point G to itself: $k \cdot G = G + G + \cdots + G$ (k times). Fig. 2 plots the real points of a representative curve of this form to illustrate the group law geometrically.

The security of ECC rests on the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP): given the base point G and a point $Q = d \cdot G$, computing the scalar d is believed to require time exponential in the bit-length of the curve's order for well-chosen curves, with respect to the best algorithms currently known. This asymmetry easy to compute Q from d, hard to invert lets ECC bind a private key to a publicly verifiable value without disclosing the key itself. In this work, the elliptic-curve key pair is generated as follows.

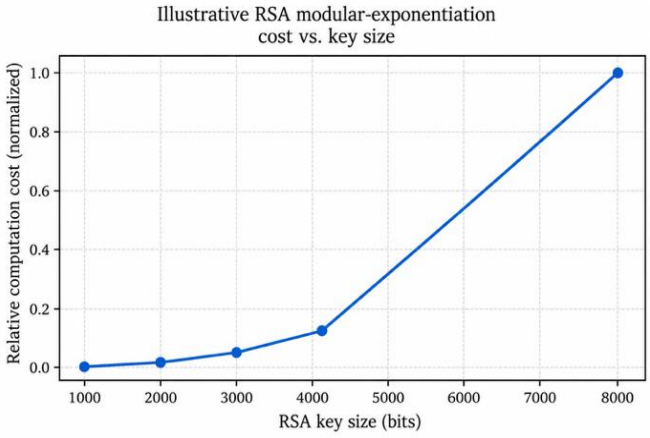


Fig. 1. Illustrative relative computation cost of RSA modular exponentiation as key size grows (naive-multiplication asymptotic model; not a measured benchmark).

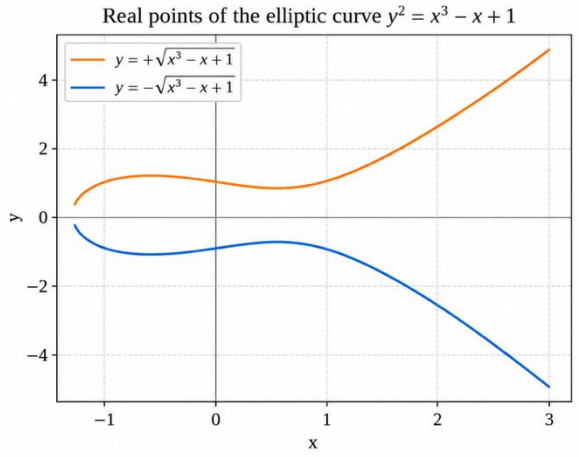


Fig. 2. Real-plane plot of the elliptic curve $y^2 = x^3 - x + 1$ (illustrative example of the group law; not the deployment curve, which is secp256k1).

Private key (d): a randomly chosen integer in the interval [1, n − 1], where n is the order of the base point G.

Public key (Q): computed as Q = d·G, a point on the curve.

The Elliptic Curve Integrated Encryption Scheme (ECIES) is a widely used hybrid construction that combines an ephemeral ECC key-agreement step with symmetric encryption and authentication. It is introduced here only as background context for the general pattern of ephemeral-key-plus-transport constructions; the protocol proposed in Section V is an ElGamal-based (ECEG) construction, not an implementation of ECIES itself. Fig. 3 compares the same RSA/ECC asymptotic costs at equal operand length, complementing the key-size-equivalent comparison of Fig. 4.

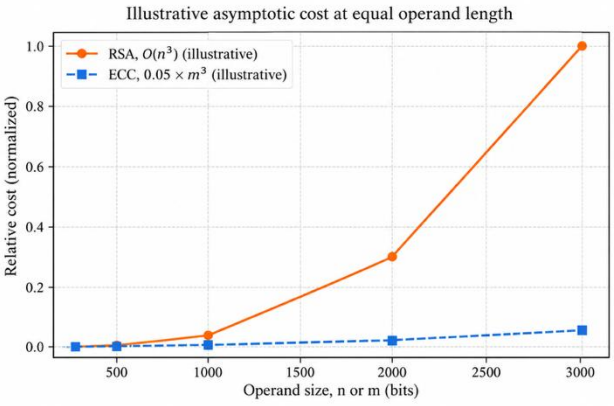


Fig. 3. Illustrative asymptotic cost of RSA and ECC at equal operand length (naive-multiplication model).

Encryption is performed by the sender using the recipient's public key Q, as follows.

- The sender generates an ephemeral private key r and the corresponding ephemeral public key R = r·G.
- The sender computes the shared point S = r·Q and derives a shared secret byte string from it, typically the x-coordinate of S.
- A key derivation function (KDF) is applied to the shared secret to produce symmetric keys: K_{enc} for encryption and K_{mac} for integrity.
- The plaintext is encrypted with an authenticated symmetric cipher, such as AES-GCM or AES-CBC combined with HMAC, using K_{enc}.
- A message authentication code is computed over the ciphertext and any associated header data using K_{mac}.
- The tuple (R, ciphertext, MAC) is transmitted to the recipient.

Decryption, performed by the recipient using the private key d, mirrors this process.

- The recipient computes the shared point S′ = d·R.
- K_{enc} and K_{mac} are derived from S′ using the same KDF and parameters used by the sender.
- The MAC is verified; the protocol aborts if verification fails.
- The ciphertext is decrypted with K_{enc} to recover the plaintext.

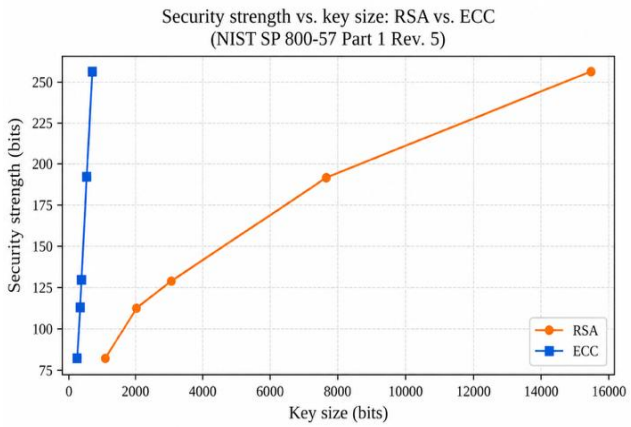


Fig. 4. Security strength versus key size for RSA and ECC, following NIST SP 800-57 Part 1 Rev. 5 [31].

As Fig. 4 illustrates, the security level obtainable per bit of key length is markedly higher for ECC than for RSA: a 256-bit ECC key (curve P-256, used here only as the NIST reference point for security-strength equivalence – the proposed protocol itself uses secp256k1, Section V-A) offers a security level comparable to a 3072-bit RSA key, both corresponding to a 128-bit security strength. This disparity widens further at higher security levels: at the 256-bit security level, a curve with a field size of 512 bits or more (e.g., P-521) is comparable to an RSA key of roughly 15360 bits, which is the central efficiency argument for adopting ECC on the resource-constrained devices considered in this study, and is analyzed quantitatively in Section VII. This comparison is a purely classical one: it assumes an adversary restricted to conventional computation. Section IX revisits the comparison under a quantum adversary, where the picture changes substantially, and situates both curves against NIST's current post-quantum migration guidance [32].

IV. Threat Model and Security Requirements

The proposed framework is evaluated under a Dolev–Yao-style adversary model [33], in which the attacker fully controls the communication channel: messages can be observed, intercepted, modified, replayed, or forged, but the attacker cannot break the underlying cryptographic hard problem; that is, the attacker cannot efficiently solve the ECDLP for the deployed curve parameters, nor invert the hash function used in the protocol. The attacker does not have access to the long-term private keys of legitimate parties and cannot read a device's protected key storage. This classical Dolev–Yao adversary is consistent with the attack surface reported in a recent systematic review of Web 3.0 cybersecurity incidents [34], which documents network-level interception, replay, and impersonation alongside Sybil attacks and phishing as recurring threats to decentralized systems. A quantum-capable adversary, who could in principle solve the ECDLP efficiently, is outside this classical threat model and is treated separately in Section IX. Under the classical model, the framework is designed to satisfy the following security requirements, which also structure the analysis in Section VI.

1) Mutual authentication: Both communicating entities should be able to verify each other's identity before any application data is exchanged.

2) Confidentiality: The character mapping and any plaintext exchanged should be unreadable to any party other than the intended recipient.

3) Integrity: any modification of the exchanged parameters or of the ciphertext during transmission must be detectable.

4) Resistance to replay: protocol messages captured in a previous session should not be reusable to impersonate a legitimate party in a new session.

5) Resistance to man-in-the-middle attacks: an adversary positioned between the two parties should not be able to establish independent, undetected sessions with each, provided the parties' long-term public keys are authentically bound to their claimed identities (Section VI-B); absent that binding, this requirement reduces to key-possession authentication rather than identity authentication (Section VI-B).

6) Pseudonymity and unlinkability: the protocol should not leak information that lets an observer link protocol messages to a real-world identity beyond what the parties intentionally disclose (pseudonymity), and, more strongly, should not let an observer tell whether two sessions involve the same party (unlinkability). As Section VI-E makes precise, this protocol targets pseudonymity by design — protocol messages carry curve points and digests, not identity attributes — but does not by itself provide unlinkability, since the long-term public keys Q_A and Q_B are reused, and hence observable, across sessions.

7) Ephemeral-session freshness: the encoding parameters used to authenticate each session should be freshly generated and distinct from those of any other session, rather than static, reusable secrets.

8) Forward secrecy: later compromise of a party's long-term private key should not allow an adversary to recover the encoding parameters, and hence the message-key material, of previously completed sessions.

9) Public-key-to-identity binding: because the protocol targets parties with no common certificate authority, an explicit external mechanism (Section VI-B) must bind each long-term public key to its claimed identity; absent that binding, mutual authentication establishes only that a peer holds a particular private key, not that the key belongs to a particular real-world or wallet identity.

Sections IV and X already draw several of the distinctions above individually; Table III collects them into a single in-scope/out-of-scope boundary for the Web 3.0-specific adversary conditions raised in the discussion of related work (Section II).

TABLE III. In-Scope and Out-of-Scope Adversary Capabilities

Adversary Class	Treatment in This Study
Network (Dolev–Yao)	In scope: full control of the channel (intercept, modify, replay, inject, block); cannot break ECDLP or invert the hash (Section IV).
Endpoint/wallet compromise (malicious wallet software, key exfiltration)	Out of scope for Sections VI-A – VI-D; partially mitigated by explicit ephemeral-key erasure (Section VI-D) but not modeled as an active capability.
Signing-request substitution / malicious dApps	Out of scope; assumed handled by wallet-level user-interface security, not by this protocol.
Phishing / social engineering	Out of scope (Section X).
Smart-contract vulnerabilities (reentrancy, logic errors)	Out of scope; an application-layer concern distinct from the authentication protocol.
Public blockchain linkability of long-term keys	Partially in scope: acknowledged as a residual pseudonymity limitation (Section VI-E), not defended against as a network-adversary capability.
Quantum adversary (Shor's algorithm)	Out of scope for the classical analysis of Sections IV - VIII; treated separately in Section IX.

V. Proposed Authentication Framework

The main operation in the proposed framework is elliptic-curve scalar multiplication. In a decentralized Web 3.0 network, two entities must authenticate each other before any data exchange takes place, and the authentication step itself must be resilient to active attacks such as man-in-the-middle interception and eavesdropping-based replay. The framework presented below is designed for constrained devices; whether it is lightweight in practice remains an empirical question requiring implementation-level benchmarking.

A. System Parameters and Key Generation

The proposed framework targets secp256k1 as its deployment curve, consistent with its use throughout the Web3/blockchain wallet ecosystem discussed in Section IX; the standardized domain parameters for secp256k1 are used directly rather than being restated here. In general, the elliptic curve E over a finite prime field is defined by the domain parameters T = (p, a, b, G, n, h), where p describes the underlying finite field, a and b define the shape of the curve, $G = (x_G, y_G)$ is the base

point, n is the order of G (the smallest positive integer such that $n \cdot G$ is the point at infinity), and h is the cofactor, $h = \#E(\mathbb{F}p)/n$, where $\#E(\mathbb{F}p)$ is the total number of points on E over $\mathbb{F}p$. The curve used in this work is cubic and is given by

$$y^2 + a_1xy + a_3y \equiv x^3 + a_2x^2 + a_4x + a_6 \pmod p \quad (1)$$

For $p > 3$ prime, the curve in Weierstrass form is

$$y^2 \equiv x^3 + ax + b \pmod p \quad (2)$$

where, $a, b \in \mathbb{F}p$ and $4a^3 + 27b^2 \neq 0$. This condition ensures that the curve is nonsingular. Each participant generates a private key k_A as a random integer and computes the corresponding public key $Q_A = k_A \cdot G$. The other participant similarly generates k_B and $Q_B = k_B \cdot G$, and the two public keys are exchanged over the (possibly adversarial) channel.

In addition to their long-term key pairs, B generates a fresh, single-use ephemeral key pair for each session: a scalar r_b and the corresponding point $R_b = r_b \cdot G$. R_b is published alongside Q_B during this same parameter-exchange step (its binding to B is subject to the same external-identity assumption as Q_B itself, Section VI-B). This per-session ephemeral point, rather than the long-term Q_B, is what the ECEG transport of Section V-C encrypts to, which is what gives the protocol forward secrecy (Section VI-D); B discards r_b once it has been used to recover a single message, so a later compromise of B's long-term key k_B does not expose past sessions' encoding parameters. In an interactive deployment, B generates R_b on demand for each new session; in a deployment where B must be able to receive a first message while offline, B instead publishes a small batch of one-time R_b values in advance and discards each one after its first use, following the same one-time-prekey pattern used in Signal's X3DH handshake.

This study adopts Schnorr signatures over secp256k1 (BIP-340 [44]) for $\mathrm{Sign}_k(\cdot)$ in Section V-C, rather than leaving the signature primitive unspecified: SHA-256 as the hash function, tagged hashes for domain separation between the different hash uses in the protocol, and a 64-byte fixed-size signature encoding. Nonces are generated deterministically from the signing key and message following BIP-340's synthetic-nonce procedure [44] (optionally mixed with fresh auxiliary randomness for side-channel hardening), the same deterministic-nonce principle standardized for ECDSA in RFC 6979 [45], which removes the catastrophic key-exposure risk that a biased or repeated per-signature random nonce creates in ECDSA-style schemes, and sidesteps ECDSA's $(r, -s)$ malleability without requiring a separate canonical-encoding rule. All received points — Q_A, Q_B, R_b, and any signature-carried commitment — are validated on receipt: each is checked to satisfy the curve equation, to differ from the point at infinity, and to lie in the correct order-n subgroup, following standard curve-point validation practice (SEC 1 [46]; NIST SP 800-186 [47]), before being used in any further computation.

B. Ephemeral ECEG Character Encoding

Before encryption, the plaintext message is processed using an Elliptic Curve ElGamal (ECEG) style encoding, extending the ephemeral mapping approach of [2]. Rather than exchanging a fixed lookup table between the 8-bit byte alphabet and points on the curve, each character is mapped, character by character, to a curve point using the scalar-multiplication relation

$$E_k = (\alpha \cdot k + \beta) \cdot G \quad (3)$$

where, k denotes the byte value (0–255) of the character being encoded, and α and β are two session-specific byte values jointly agreed upon by the sender and receiver during the encoding phase described below. Requiring a raw curve-point coordinate to already fall in (0, 255) is not usable on a 256-bit curve: both coordinates of a random point land in that range with probability on the order of 2^{-496}, making rejection sampling computationally infeasible. The sender therefore derives α and β from an ephemeral point with a hash-based key-derivation step, then re-encodes them as a curve point before transport (Section V-C requires a curve point, not an arbitrary integer pair), as follows.

Step 1: pick a random integer i.

Step 2: compute the ephemeral elliptic-curve point

$S = i \cdot G = (X_S, Y_S)$, where G is the generator point of the selected curve.

Step 3: derive the encoding parameters with a key-derivation function applied to the point's coordinates: $(\alpha, \beta) = \mathrm{Trunc}_{16}(\mathrm{hash}(X_S \,\|\, Y_S))$, i.e., take the first 16 bits of $\mathrm{hash}(X_S \,\|\, Y_S)$ and split them into two bytes α and β. If $\alpha = 0$, the sender discards the derived pair and regenerates i until $\alpha \neq 0$, ensuring that the byte-to-point mapping remains injective over the 256-byte alphabet.

Step 4: combine the two bytes into a single 16-bit integer $m = 256 \cdot \alpha + \beta$, and encode m as a curve point $M = \mathrm{EncodeToPoint}(m)$ using the integer-to-point technique of Koblitz [41]: for $j = 0, 1, 2, \ldots$, form the candidate x-coordinate $x = m \cdot K + j$ for a fixed public padding constant K chosen so that $m \cdot K$ stays far below p, and stop at the first j for which $x^3 + ax + b$ is a quadratic residue mod p, setting y to a square root of $x^3 + ax + b \bmod p$. Decoding reverses this: recover $m = \lfloor x / K \rfloor$ from M's x-coordinate, then split it back into $\alpha = \lfloor m / 256 \rfloor$ and $\beta = m \bmod 256$. Because roughly half of the field elements are quadratic residues, encoding succeeds within the first few values of j with overwhelming probability.

Only M – not α and β directly – is ever placed on the curve or on the channel; α and β themselves remain ordinary 16-bit integers used solely to build M, to form the signature input used in Section V-C, and to build the byte-to-curve-point mapping table. Because that reduction is only 16 bits wide, α and β are not by themselves a cryptographically strong secret – at that size they would be brute-forceable if ever exposed on their own. Their practical protection instead comes from the fact that they are only ever transported inside the authenticated ECEG ciphertext M of Section V-C, never sent in the clear; this trade-off is stated here explicitly rather than left implicit.

C. Signature- and Two-Ephemeral-Key ECEG-Based Mutual Authentication

Once α and β are generated and encoded as the curve point M of Section V-B, they must be transported to B and confirmed by both parties before either side uses them, without ever placing

(α, β) or M on the channel unencrypted, and with each party's authorship of its own messages bound to its long-term key pair rather than to any state carried over from a previous session. Every symbol below is defined once and used consistently with Fig. 5.

A picks a fresh ephemeral scalar r and forms the ECEG transport of M under B's per-session ephemeral point R_b (Section V-A) rather than under B's long-term Q_B: $C_1 = r \cdot G$, $C_2 = M + r \cdot R\_b$, $C = (C_1, C_2)$. A also generates a fresh, single-use session nonce N_a and computes a signature $\sigma\_a = \mathrm{Sign}_{k_A}(\mathrm{hash}(C \parallel N\_a \parallel Q_A \parallel Q_B \parallel R\_b))$ under its own long-term private key — binding the specific R_b used into the signed transcript so that a later substitution of R_b during the key-exchange step cannot go undetected — then sends (C, N_a, σ_a) to B. B first checks that N_a has not been seen before, rejecting the message as a replay if it has, and verifies σ_a against Q_A; if either check fails, B aborts before doing any further work on C, in particular before performing the ECEG decoding.

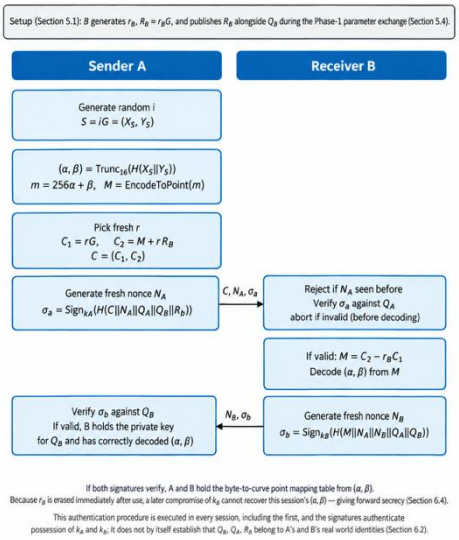


Fig. 5. Signature- and nonce-based mutual authentication and encoding parameter transport between sender A and receiver B.

If both checks succeed, B recovers $M = C_2 - r\_b \cdot C_1$ using the ephemeral scalar r_b generated for this session (Section V-I), then immediately erases r_b — the step that gives the protocol forward secrecy (Section VI-D), since nothing derivable from B's long-term key k_B can later reconstruct r_b or M — and decodes (α, β) from M as in Section V-B. B then generates its own fresh nonce N_b and computes $\sigma\_b = \mathrm{Sign}_{k_B}(\mathrm{hash}(M \parallel N\_a \parallel N\_b \parallel Q_A \parallel Q_B))$, and sends (N_b, σ_b) to A. A verifies σ_b against Q_B; if verification succeeds, A concludes that whoever holds the private key for Q_B received and correctly decoded (α, β), and both sides finalize the byte-to-curve-point mapping table used for subsequent character-by-character encryption.

This design authenticates each party as the holder of the long-term private key corresponding to its public key, in every session including the first: unlike a design that relies on shared history from a previous session, a valid signature cannot be produced without k_A or k_B regardless of whether A and B have interacted before, so it does not have the bootstrapping gap that a history-dependent check would have on a first encounter. It does not by itself establish that Q_A or Q_B belongs to the real-world identity it is claimed to represent; that binding is the external identity layer discussed in Section VI-B, but it does rule out an adversary who has not compromised either long-term key from completing the handshake under a false claim to already hold one of these keys. Freshness and replay-resistance rest on the per-session nonces N_a, N_b, which B and A each check against a short-lived record of recently used values, rather than on synchronized state carried over from a previous session; this also avoids the resynchronization problems (e.g., after a device restart or an interrupted session) that a previous-session-dependent design would otherwise create.

Both nonces are generated as 128-bit values from a cryptographically secure random-number generator, giving a collision probability across k sessions on the order of $k(k-1)/2 \times 2^{-128}$ by the birthday bound — negligible for any realistic session count. Each party retains a replay cache of nonces it has already seen and evicts each entry a fixed window after first recording it (tens of seconds to a few minutes, matching the round-trip tolerance conventionally used for challenge–response freshness, e.g., in Kerberos), so the cache size stays bounded by the message-arrival rate rather than growing without bound over the life of a long-lived identity; a replayed message that arrives after its originating entry has already been evicted is accepted as if fresh only if the underlying nonce happens to collide with a genuinely new one, which the birthday bound above rules out. After a restart, a party simply begins from an empty cache; a validator or wallet backend spanning multiple nodes shares this cache (or replicates it with bounded staleness) across nodes so that a replay accepted at one node is rejected at the others.

D. Protocol Execution

The protocol runs in three phases, as shown in Fig. 6. First, the shared elliptic-curve domain parameters are agreed upon by sender A and receiver B; long-term public keys Q_A and Q_B are exchanged, and B additionally generates and publishes a fresh, session-specific ephemeral point R_b (Section V-A); the binding of Q_A, Q_B, and R_b to A's and B's real-world identities is treated as an external assumption rather than established by this exchange itself, as discussed in Section VI-B. Second, the ephemeral encoding procedure of Section V-B is performed to produce α and β, which are transported under R_b rather than under B's long-term key and mutually confirmed using the signature- and nonce-based handshake of Section V-C; this step proves that each party holds the private key for its claimed long-term public key, in addition to confirming that B correctly recovered (α, β), and at the end of this stage both parties share the same session-specific encoding parameters, B has erased r_b (Section VI-D), and neither has exchanged the full byte-to-curve-point mapping table itself. Third, A encrypts the message character by character with the derived mapping and sends it to B as ciphertext; B has already recovered and decoded (α, β) in the previous phase and uses the resulting byte-to-curve-point mapping table to recover the original plaintext character by character.

This design reduces the risk of replay attacks, since authentication in each session depends on encoding parameters that combine fresh randomness with values distinct from any previous session; resistance to man-in-the-middle substitution

additionally requires the public-key binding discussed in Section VI-B. Even if an adversary intercepts the ciphertext components during transmission, recovering B's private key k_B, and, separately, recovering B's per-session ephemeral scalar r_b, from the intercepted values is computationally infeasible under the ECDLP hardness assumption discussed in Section VI, since the ephemeral points i·G, R_b = r_b·G, and the ciphertext component $C_1 = r \cdot G$ reveal only curve points, not the underlying scalars i, r, r_b, or k_B. The session's encoding attributes α and β are a different case, addressed in Section VI-A: at only 16 bits, they are not ECDLP-hard on their own, and are protected instead by never being sent except inside the encrypted transport M of Section V-C.

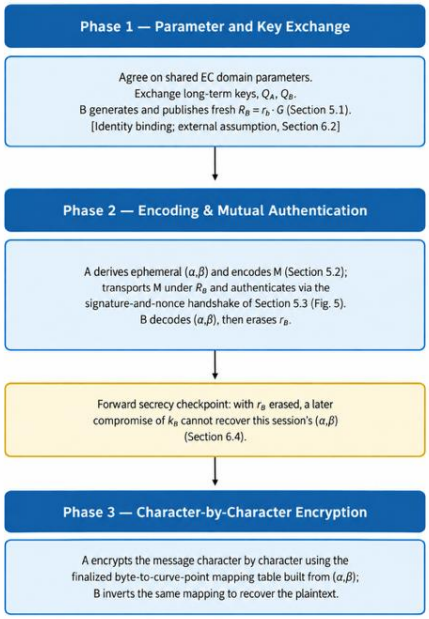


Fig. 6. End-to-end execution flow of the proposed ECEG authentication framework, including per-session signature verification.

VI. Security Analysis

This section gives an informal, assumption-based analysis of the protocol with respect to each requirement identified in Section IV. The analysis rests on the Dolev–Yao adversary model [33] and on the standard assumptions that the ECDLP is computationally hard for the chosen curve parameters and that the ECEG transport is IND-CPA secure under the relevant group assumption (e.g., DDH), while the hash function used in the signature inputs is preimage- and collision-resistant. Section VI-F then specifies a concrete plan for replacing this informal argument with a machine-checked one.

A. Resistance to Eavesdropping

An adversary eavesdropping on the channel sees only public curve points – e.g., Q_A, Q_B, ciphertext components, and hash/signature outputs. Recovering the long-term or ephemeral private scalars k_A, k_B, r_b, or i from these observations requires solving the ECDLP, which is assumed computationally infeasible for curves of appropriate size. The 16-bit pair (α, β) is a different matter: at only 2^{16} possible values it is not itself ECDLP-hard, and would be recoverable by exhaustive search from even a single known plaintext/ciphertext pair under a fixed encryption key. Its confidentiality therefore rests entirely on the semantic security (IND-CPA) of the ECEG transport that carries it — specifically, on the pair (r, R_b) never repeating across sessions (Sections V-A, V-C) — rather than on the small parameter space itself being hard to search; framed differently, the 16-bit encoding provides no obfuscation-level security on its own and should not be read as doing so. Because r and R_b are both freshly generated per session and never reused, each transport instance is independently randomized, so an eavesdropper who cannot invert the transport does not obtain (α, β) directly, and passive monitoring does not reveal the mapping used for the actual message.

B. Resistance to Man-in-the-Middle Attacks

The signature step of Section V-C means a man-in-the-middle adversary cannot forge a valid session on behalf of A or B without the corresponding long-term private key k_A or k_B: substituting tampered values for C, N_a, or N_b causes signature verification to fail, and the session aborts before either party acts on the forged data. What this check cannot do is establish that Q_A or Q_B belongs to the real-world identity it is claimed to represent in the first place. If an adversary presents its own key pair as Q_B at the very start of the exchange – before any signature has been exchanged – and A has no independent way to confirm that this Q_B is genuinely B's, A can complete a validly signed handshake with the adversary under the adversary's own key. This residual gap is a binding problem between identity and public key, not a weakness in the signature or key-transport steps themselves; closing it requires the external identity layer described next.

A concrete instantiation of this external identity layer, rather than an abstract placeholder, is available using the W3C Decentralized Identifier (DID) Core [42] and Verifiable Credential Data Model [43] standards, and integrates directly into the parameter exchange of Section V-D rather than sitting alongside it as a separate system. Each party publishes its long-term public key as the verification method of a DID document — for instance, under the did: key method, which encodes the public key directly in the identifier and needs no registry lookup, or under a ledger-anchored method such as did:ethr when linkage to an existing wallet address is required. A credential issuer that the verifying party already trusts (a prior counterparty, an organization, or any other chosen trust anchor) signs a verifiable credential attesting that a given DID controls a given real-world or wallet identity. During the exchange of Q_A, Q_B, and R_b in Section V-D's first phase, each party additionally presents its DID document and this credential; the peer checks the credential's issuer signature and confirms that the DID document's verification method matches the presented public key before proceeding to the signature-and-key-transport steps of Section V-C. This binds the specific public key used later in the handshake to a checked identity claim up front, rather than accepting whichever key arrives first, closing the substitution gap described above; the pattern of folding a DID/VC presentation directly into an existing handshake, rather than treating it as a separate protocol layer, follows the approach recently described for integrating DIDs and verifiable credentials into the TLS handshake [50].

This mechanism does not remove every trust assumption — it shifts the question from "is this public key really B's" to "is

this credential issuer trustworthy and is its signature currently valid," the same reduction any certificate- or credential-based scheme makes — but the resulting trust anchor is a credential issuer or prior relationship chosen by the parties themselves rather than a centralized certificate authority, consistent with the self-sovereign-identity setting motivating this study (Section I). Formally analyzing the combined system — the DID/VC presentation step together with the signature-and-key-transport handshake of Section V-C, as a single verified protocol rather than as two independently analyzed components — is identified as follow-up work in Section 10.

C. Resistance to Replay Attacks

Each session includes a fresh, single-use nonce (N_a from A, N_b from B, Section V-C), and both the signature and the confirmation checks are computed over values that include these nonces. An adversary who records a complete protocol transcript from one session and replays it in a later session presents a nonce that the recipient has already seen, so B's nonce check (or A's, for the return message) rejects it before any signature verification or ECEG decoding is attempted. Because this check happens first, a replayed message is cheap to reject; a flood of freshly-constructed but invalidly-signed messages is more expensive, since it still costs a signature verification per message, and defending against that kind of resource-exhaustion load is outside the assumption-based analysis in this section – rate limiting or a lightweight pre-filter would be needed in a production deployment (Section X).

D. Forward Secrecy and Key Compromise

Because α, β, the ephemeral scalar i used to derive them, and the per-session ephemeral scalars r and r_b (Section V-A, V-C) are all generated freshly for each session, none of them is a deterministic function of the long-term private keys k_A or k_B alone, so a device that is captured while offline does not thereby leak the encoding parameters of past sessions from its stored long-term key material. Unlike the static-key ECEG transport used in [3] and in an earlier draft of this protocol, the transport of Section V-C encrypts (α, β) under B's per-session ephemeral point $R_b = r_b \cdot G$ rather than under B's static Q_B, and B erases r_b immediately after using it to recover a single message. Consequently, an adversary who has recorded C_1 and C_2 from a past session and later obtains B's long-term private key k_B gains no advantage in reconstructing that session's r_b, since r_b is chosen independently of k_B and is not recoverable from R_b under the ECDLP hardness assumption; without r_b, the adversary cannot compute $r_b \cdot C_1$ and so cannot recover (α, β) from the recorded transcript. The protocol therefore provides forward secrecy against later compromise of a long-term private key, conditional on B genuinely erasing r_b after each session's single use — an explicit key-erasure assumption stated here rather than left implicit, and one that a production implementation must enforce (e.g., by zeroing the scalar's memory immediately after the single point multiplication that consumes it) for the property to hold in practice. This conditional guarantee, together with its formal counterpart in the verification plan of Section VI-F, replaces the unconditional absence of forward secrecy that an earlier, static-key version of this design would have inherited from [3]. The same forward-secrecy guarantee also requires A to erase its session-specific scalars i and r after the session completes; retaining either scalar would allow recovery of past session material from recorded data. Accordingly, the erasure condition is a protocol-state requirement for both parties, not only for B.

E. Anonymity and Residual Risks

The protocol limits what an on-path observer can learn about the identities involved in a given exchange, since the exchanged values are curve points and hash digests rather than identity attributes. Full anonymity in the sense used by self-sovereign identity systems, however, also depends on how public keys are bound to real-world identities at the application layer, for instance, through the decentralized-identifier/verifiable-credential mechanism integrated into the exchange in Section VI-B, which establishes identity binding but does not by itself change what an observer of the resulting public keys can infer. This distinction matters specifically for the Web 3.0 setting targeted here (Section II-D): unlike a provisioned IoT device with a fixed operator-known identity, a Web 3.0 wallet's long-term public key may itself be linkable across sessions through blockchain analysis even if this protocol's ephemeral parameters are not. Reasoning about that residual linkability is an application layer concern outside this study's scope, but it should be addressed jointly with the SSI layer in any deployment. The analysis in this section is informal rather than a machine-checked proof; Section VI-F identifies automated protocol verification as an important next step before the protocol is considered ready for production deployment.

F. Formal Verification Specification and Reproducibility Artifact

Sections VI-A – VI-E give an informal, assumption-based argument. To move beyond that argument, this subsection specifies a concrete formal-verification model and its required security queries, precisely enough to be encoded in Tamarin [54] as a supplementary reproducibility artifact without further design decisions. No such encoding has yet been produced or executed, and accordingly no machine-checked verification result is claimed in this study; preparing and running that encoding is identified as the immediate next step in Section X.

1) Adversary model: The standard Dolev–Yao network adversary of Section IV, in full control of the network and able to intercept, block, modify, replay, and inject messages, and able to compromise any number of protocol sessions but not the internal state of an honest party during a session, nor the long-term private keys k_A, k_B.

2) Tool and primitives: The specification below abstracts the two-ephemeral-key ECEG transport using a symbolic shared-point relation, rather than attempting to encode concrete secp256k1 point arithmetic. It treats the hash function as one-way and digital signatures through the verifier's signature primitives. Because the protocol no longer includes an XOR operation, the resulting model would not require XOR equational reasoning. Tamarin [54] is used as the reproducibility target; Scyther [25] remains relevant as the tool used to verify the closest prior art [3].

3) Protocol encoding: The roles of A and B are encoded as two processes exchanging exactly the messages of Section V-D, with fresh session-specific values (i, r, r_b) declared using

the tool's freshness construct so that each session uses independent randomness, while the long-term private keys k_A, k_B remain persistent across sessions, and with r_b scoped so the process has no further access to it once it has been used for a single decryption, modeling the explicit erasure required by Section VI-D.

Queries.

- Secrecy of α, β, r_b, k_A, and k_B: the adversary process must never derive these values, regardless of how many sessions run concurrently.
- Injective agreement (correspondence) between A and B on the tuple of session-specific values (α, β, R_b, and the session identifier), so that every session in which A completes with B corresponds to a distinct completed session of B with A — the standard formalization of both mutual authentication and replay resistance, with R_b included so that a substitution of the ephemeral point between phases is itself caught as a disagreement.
- Forward secrecy under long-term key compromise: secrecy of α and β from an already-completed session even after both parties' long-term private keys are revealed to the adversary once the session has ended, directly encoding the erasure-conditional guarantee argued informally in Section VI-D, rather than treated as an optional strengthening as it would have been for the earlier static-key design.
- Expected deliverable: A machine-checked verdict verified, or a concrete attack trace for each query above, together with the tool version and the model source, to be released as a supplementary artifact. Executing this plan and reporting its outcome, whichever it is, is identified in Section X as the immediate next step before the protocol can be considered validated beyond the assumption-based argument given in Sections VI-A – VI-E.

G. Security-Claim Summary

Table IV records, for each property discussed in Sections 6.1–6.6, the strongest claim this study makes and the assumption or limitation on which it rests. It is intentionally conservative: properties that depend on external identity binding or on an unexecuted validation step are marked conditional or not claimed rather than folded into an unqualified security statement.

TABLE IV. SECURITY-CLAIM SUMMARY FOR THE PROTOCOL OF SECTION V

Property	Status	Basis and limitation
Key-possession authentication	Claimed	Schnorr signatures over the current transcript (Section V-C); assumes uncompromised long-term keys and point validation (Section V-A).
Identity authentication	Conditional	Holds only after the DID/VC binding check of Section VI-B; depends on the chosen credential issuer / DID method.
MITM resistance	Conditional	Signatures block tampering after key selection; initial key substitution is prevented only by the identity binding of Section VI-B.
Replay resistance	Claimed	Fresh 128-bit nonces checked against a bounded replay cache (Section V-C); requires the stated cache window.
Confidentiality of (α,β)	Claimed	Rests on IND-CPA security of the ECEG transport with fresh (r, R_b) per session (Section VI-A); the 16-bit values are not secrets in themselves.
Forward secrecy	Conditional	Requires erasure of i and r by A and r_b by B after the session (Section VI-D).
Pseudonymity	Claimed	Messages carry curve points and digests, not identity attributes (Section VI-E).
Unlinkability/anonymity	Not claimed	Long-term keys, DIDs, and wallet addresses remain linkable across sessions (Sections IV, VI-E).
Machine-checked verification	Not claimed	Verification plan specified (Section VI-F); no verifier run reported in this study.
Empirical lightweight performance	Not claimed	Analytical operation count only (Table VI); no latency, energy, memory, or throughput measurements performed.
Post-quantum security	Not claimed	Classical ECC only; migration path outlined in Section IX.

VII. COMPUTATIONAL COMPLEXITY ANALYSIS

The efficiency analysis in this section separates generic RSA/ECC complexity from the protocol-level costs of the proposed scheme. The generic comparison is intended to show the security-to-key-size relationship; it is not a substitute for implementation-level benchmarking.

A. RSA Modular Exponentiation

The main computation of RSA is the modular exponentiation $c = m^e \bmod N$. Its cost under repeated squaring depends on n, the bit length of the modulus N (an upper bound on the number of modular multiplications), and M(n), the cost of a single big-integer multiplication or modular reduction. The total cost of modular exponentiation is therefore roughly $O(M(n)\cdot n)$. School-book multiplication gives $M(n) = O(n^2)$ and hence a rough bound of $O(n^3)$; more advanced algorithms such as Karatsuba or FFT-based multiplication improve the constant, and the exponent on M(n), but the dependence on n remains large, and n must typically be 2048 bits or more to reach an acceptable security level.

B. ECC Scalar Multiplication

The main computation in ECC is scalar multiplication $Q = k\cdot P$, realized through repeated point additions and doublings. Its cost depends on m, the bit length of the field elements used to

represent coordinates (the number of point operations is proportional to m), and M(m), the cost of one field multiplication. This gives a total scalar-multiplication cost of order O(M(m)·m), a bound of similar form to O(m³) under naive field multiplication. The key difference is that m can be substantially smaller than n while still achieving the same security level, as Table V shows, which reduces key representation and can reduce the cost of corresponding public-key operations; actual runtime and energy benefits depend on implementation, hardware, and library choices and are not measured here. Values follow NIST SP 800-57 Part 1 Revision 5 [31].

TABLE V. COMPARABLE SECURITY LEVEL: ECC KEY SIZE VERSUS RSA KEY SIZE

Security Level (bits)	ECC Field Size (bits)	RSA Key Size (bits)
256	512+	15,360
192	384–511	7,680
128	256–383	3,072
112	224–255	2,048
80	160–223	1,024

As Table V and Fig. 7 show, the RSA key size is 3072 bits for a 128-bit security level, while the corresponding ECC key size is only 256 to 383 bits. The difference widens further at higher security levels: a 256-bit security level requires an RSA key of roughly 15,360 bits, versus an ECC key of 512 bits or more. Under the naive-multiplication model, both schemes have cost growing with the cube of the key length, but the ECC key length required grows much more slowly than the RSA key length required as the target security level increases. This supports the use of ECC where storage and communication overhead matter, but it does not establish lower end-to-end runtime for the complete protocol; that requires empirical measurement.

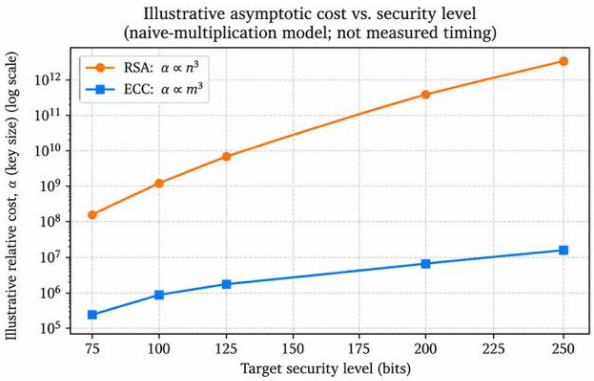


Fig. 7. Illustrative asymptotic cost of RSA and ECC as a function of target security level.

C. Protocol-Level Operation Count and Communication Cost

The comparison in Sections VII-A – VII-B evaluates generic RSA and ECC operations rather than the protocol proposed in Section V itself. Table VI instead counts, directly from the specification of Sections V-A – V-D, every EC scalar multiplication, signature operation, hash evaluation, message, and byte of state that one completed session requires of each party.

TABLE VI. PER-SESSION OPERATION COUNT FOR THE PROTOCOL OF SECTION V

Operation (per completed session)	A	B
EC scalar multiplications, total (handshake, incl. Schnorr)	6	5
Byte-to-point table construction (one-time per session)	256	256
Signature generations	1	1
Signature verifications	1	1
Explicit SHA-256 hash/KDF evaluations (protocol)	2	1
Messages sent (handshake only)	1	1
Bytes transmitted (handshake only)	≈ 146 B	≈ 80 B
Long-term key state	k_A + Q_B	k_B + replay cache

The scalar-multiplication total breaks down as follows: for A, i·G (encoding, Section V-B) + r·G + r·R_b (Section V-C) + 1 scalar multiplication for generating σ_A + 2 for verifying σ_B (Schnorr verification counted conservatively as two scalar multiplications in the unbatched verification equation) = 6. For B, r_b·G (Section V-A) + r_b·C₁ (Section V-C) + 1 scalar multiplication for generating σ_B + 2 for verifying σ_A = 5. These totals therefore include the elliptic-curve work performed inside the Schnorr signature operations; the signature-generation and signature-verification columns in Table VI report those operations separately for clarity. An implementation using multi-scalar-multiplication techniques can reduce verification cost. The 256-entry byte-to-point table is built once per session and then reused for every character of the message by table lookup, so its construction cost is independent of message length L rather than growing with it. Persistent state is dominated by the 32-byte long-term private key on each side; B additionally holds a bounded replay cache (Section V-C). Table VI's message and byte counts cover only the cryptographic authentication phase and exclude Phase-1 domain/key exchange and DID/VC presentation overhead.

A separate communication cost, not included in the handshake byte count in Table VI, is the bandwidth expansion of the character-level ECEG encryption in Section V-D's third phase: each plaintext byte is represented on the wire as a full secp256k1 curve point, 33 bytes compressed or 65 bytes uncompressed, a 33× to 65× expansion over the one byte of plaintext it carries. A conventional hybrid alternative — using the already-established per-session shared point r·R_b (Section V-C) to key a symmetric authenticated cipher such as AES-GCM or ChaCha20-Poly1305 for the bulk message, as Section III describes for ECIES — would instead add only a small constant-size tag (typically 16 bytes) regardless of message length, with no per-character point encoding. This study retains character-level ECEG encoding because it is the specific mechanism inherited from [2,3] that the threat model and verification plan of Sections IV and VI-F are built around; a deployment that prioritizes bandwidth over continuity with that

prior design should instead use the hybrid AEAD construction for the bulk message while keeping the ECEG-over-two-ephemeral-key transport of Section V-C for the (α,β) handshake itself.

Published implementation-level benchmarks corroborate the direction, though not the exact magnitude, of the efficiency argument above: Hammi et al. [48] propose a lightweight ECC-based mutual authentication scheme for IoT and report it as computationally and communication-efficient relative to prior IoT authentication schemes on constrained hardware, in the same general direction as the asymptotic argument of Sections VII-A – VII-B, though that study does not benchmark against RSA directly; Li and Hu [49] report a fresh ECC-based IoT authentication and key agreement protocol reducing computational overhead by more than 37% and communication cost by more than 5% relative to prior schemes, using a per-session operation-count methodology similar in spirit to Table VI. Neither benchmark is of the protocol proposed in this study, so they are cited as external corroboration of the general ECC-efficiency direction rather than as a substitute for the hardware benchmarking identified as future work in Section X.

VIII. Discussion

The analyses of Sections VI and VII lead to two complementary conclusions. From a security standpoint, tying authentication to ephemeral, session-unique parameters instead of static secrets, and grounding that design in the ECDLP hardness assumption, is intended to make eavesdropping and replay harder for an adversary limited to the Dolev–Yao model of Section IV, without requiring a common certificate authority; resistance to man-in-the-middle interception additionally depends on the public-key binding discussed in Section VI-B. From an efficiency standpoint, the asymptotic and key-size comparisons in Section VII support the well-known ECC security-to-key-size advantage, though actual end-to-end performance depends on implementation, hardware, and library choices not measured here.

These security and representation-size properties are complementary rather than independent: the relatively small ECC keys and points make per-session ephemeral operations practical to consider on constrained devices, while the actual cost of the proposed ECEG encoding and signature-based handshake remains an empirical question. The security properties described in Section VI therefore should not be inferred from key-size ratios alone.

As discussed in Section II-B, the framework's relationship to the closest prior art [3] is best understood as an adaptation at two levels rather than a wholesale new cryptographic mechanism: both protocols inherit the same ephemeral ECEG character-encoding step from [2], rather than a static, exchanged lookup table or a static identity-bound key exchange as in several of the RFID and smart-card schemes reviewed in Section II [4,5,9,10]. Beyond that shared encoding step, however, the two protocols diverge: [3] transports the encoding parameters under a static recipient key and confirms them with a hash-and-XOR value chained to the previous session, whereas this study authenticates each party with digital signatures over its long-term key and transports the encoding parameters under a fresh per-session ephemeral key (Section V-C), which is what closes the forward-secrecy gap identified in Table I. What this study adds on top of that protocol-level change is the explicit threat model, the pseudonymity/unlinkability analysis, the concrete DID/VC identity-binding mechanism (Section VI-B), and the verification plan needed to argue that the resulting design is appropriate specifically for Web 3.0 peer-to-peer interactions, where the two interacting parties may have no previously established trust relationship and no common certificate authority.

IX. Post-Quantum Outlook

The security analysis in Section VI and the efficiency case in Section VII both rest on a classical adversary: one that cannot efficiently solve the Elliptic Curve Discrete Logarithm Problem (ECDLP). A quantum adversary running Shor's algorithm can, in principle, solve the ECDLP in polynomial time, and 2025–2026 has seen rapid, concrete progress in estimating exactly how many quantum resources that would take for the curve sizes relevant to Web 3.0. This section situates the proposed framework against that fast-moving picture, states plainly what it implies for the protocol proposed here, and outlines a migration path consistent with current NIST guidance.

A. The Narrowing Quantum Resource Gap for ECDLP

TABLE VII. Successive 2020–2026 Estimates of the Logical Qubits Needed to Solve the 256-Bit Elliptic-Curve Discrete Logarithm Problem on a Prime-Field Curve

Source	Year	Venue / Status	Logical Qubits (256-bit curve)
Häner et al. [PQCrypto 2020]	2020	Peer-reviewed (PQCrypto 2020)	≈ 2124
[35]	2026	Peer-reviewed (PRX Quantum 2026)	< 1200–1450
[36]	2026	Peer-reviewed (EUROCRYPT 2026)	≈ 1193
[37]	2026	Preprint (arXiv)	≈ 1333
Häner et al. [51]	2026	Preprint (IACR ePrint / arXiv)	≈ 1450 (19,397 physical qubits; ~25.7 days)

Table VII summarizes successive 2026 reductions in the estimated number of logical qubits needed to solve the 256-bit ECDLP on a prime-field curve such as secp256k1, the curve family underlying most Web3 wallets and curves in the same security class as those discussed in this study. Babbush et al. [35], a Google Quantum AI, Ethereum Foundation, and Stanford collaboration, reported that Shor's algorithm for this problem can execute with either fewer than 1200 logical qubits and fewer than 90 million Toffoli gates, or fewer than 1450 logical qubits and fewer than 70 million Toffoli gates, running in minutes on fewer than half a million physical qubits at a 10^{-3} error rate. Chevignard, Fouque, and Schrottenloher [36], published in the peer-reviewed EUROCRYPT 2026 proceedings, independently reduced the estimate to roughly 1193 logical qubits, below the qubit count separately estimated for factoring 3072-bit RSA; Luo et al. [37] independently reduced the earlier estimate of roughly 2124 qubits reported by Häner et al. to about 1333. None of these results has been demonstrated on physical hardware at the required scale, but the direction is unambiguous: multiple independent groups converged on roughly the same 1200–1450-qubit range within a few months of each other in

2026, down from the prior benchmark of over 2100 qubits. A September 2026 estimate by Häner et al. [51] translates a comparable logical-qubit count into a physical-hardware figure for a trapped-ion architecture: roughly 19,397 physical qubits solving the same problem in approximately 25.7 days, down from prior trapped-ion estimates of 1.2 to 9.4 million physical qubits. As preprints or very recent results, these figures should be read as a fast-moving direction rather than a settled number; the recommendation in Section IX. C is unaffected by exactly which proves most durable.

Figures are as reported by each source; none has been demonstrated on physical quantum hardware at the scale required, and estimation methodology differs slightly across sources.

B. Implications for this Framework

The framework proposed in this study is, like essentially all currently deployed Web 3.0 authentication and wallet infrastructure, built entirely on classical ECC and therefore offers no protection against a sufficiently capable quantum adversary: such an adversary could recover a party's private key k_A or k_B from its public key, and could therefore forge the mutual-authentication handshake of Section V-C outright, independent of the ephemeral, session-unique design that protects against the classical replay and man-in-the-middle attacks analyzed in Section VI. A further, distinct risk is "harvest-now, decrypt-later": an adversary who records protocol transcripts or on-chain public keys today could decrypt or forge against them retroactively once a cryptographically relevant quantum computer exists, which matters specifically for long-lived Web 3.0 identities that are expected to remain valid for years. It is important to be precise about what current estimates do and do not show: no cryptographically relevant quantum computer capable of running these circuits exists today, and the qubit counts in Table VII, while dropping quickly, remain roughly three orders of magnitude beyond current hardware. The risk this section addresses is therefore one of migration planning, not of an immediate break.

C. Standards Guidance and a Migration Path

NIST finalized the first three post-quantum cryptography standards in August 2024: FIPS 203 (ML-KEM, a module-lattice key-encapsulation mechanism), FIPS 204 (ML-DSA, a module-lattice digital signature scheme), and FIPS 205 (SLH-DSA, a stateless hash-based digital signature scheme) [38]. NIST Internal Report 8547 (initial public draft) sets out a migration timeline in which algorithms providing 112-bit classical security, including RSA-2048 and ECC P-256, are to be deprecated after 2030 and disallowed after 2035, with the same disallowance date applying to higher-strength classical public-key algorithms including the curves discussed in this study [32]. Read against that timeline, the ECC-based framework proposed here remains an appropriate near- and medium-term choice, consistent with current NIST guidance, but it is not a long-term end state.

A practical migration path, consistent with how the wider field is approaching this problem, is a hybrid construction rather than a wholesale redesign: retaining the ephemeral ECEG encoding and signature-based handshake of Section V for their efficiency and their resistance to classical replay and man-in-the-middle attacks, while wrapping the long-term key-establishment step in a post-quantum key-encapsulation mechanism such as ML-KEM, so that an adversary must break both the classical and post-quantum components to succeed. Recent quantum-resistant self-sovereign-identity frameworks [39,40] establish precedent for this pattern by layering NIST-standardized post-quantum primitives onto a decentralized identity architecture; designing and evaluating a concrete hybrid variant of the present framework along these lines, rather than merely outlining the path as this section does, is identified as future work in Section X.

X. Limitations and Future Work

The evaluation in this study is analytic rather than empirical: the security discussion in Section VI is an informal argument based on standard cryptographic hardness assumptions rather than a machine-checked proof, and the efficiency discussion in Section VII is based on asymptotic complexity and established key-size equivalences rather than measured execution time on physical hardware. This limitation directly implies five concrete directions for follow-up work.

1) Formal verification of the protocol. The plan of Section VI-F should be encoded in Tamarin [54] and run against the documented queries for mutual authentication, replay resistance, secrecy, and erasure-conditional forward secrecy. The present manuscript reports no machine-checked verdict because that encoding has not yet been produced or executed.

2) Formal and empirical validation of the identity-binding mechanism. The DID/VC binding of Section VI-B is specified concretely but has not been analyzed as part of a single verified system together with the signature-and-key-transport handshake of Section V-C, nor implemented against a live DID method (e.g., did: key or did:ethr) to measure the latency and message-size overhead the credential-presentation step adds to the base handshake of Table VI.

3) Hardware benchmarking. To confirm the asymptotic gains of Section VII under practical operating conditions rather than theoretical bounds, the protocol should be implemented on representative resource-constrained platforms, such as microcontroller-class IoT devices or embedded wallet hardware, with measured wall-clock execution time, memory usage, and energy consumption, reported against the same curve parameters and RSA baseline used by the closest prior art [3] to allow a direct, like-for-like comparison rather than a comparison of asymptotic bounds alone.

4) Integration with anomaly detection. The authentication layer could be supplemented with a separate anomaly-detection mechanism, such as the machine-learning-based approach studied for cloud data in [27], to recognize statistically unexpected authentication patterns, for example, an unusually high number of failed handshakes even when each individual attempt is cryptographically valid.

5) Hybrid post-quantum migration. As Section IX discusses, a concrete hybrid variant of the protocol that wraps the long-term key-establishment step in a NIST-standardized post-quantum key-encapsulation mechanism such as ML-KEM

[38] should be designed and evaluated, following the precedent of recent quantum-resistant self-sovereign-identity frameworks [39,40], so that the protocol's classical efficiency advantages are retained in the near term without leaving it exposed once quantum resource estimates such as those in Table VII are realized in hardware.

The present framework addresses authentication and identity binding only. It does not address smart-contract-level vulnerabilities such as reentrancy or logic errors, nor the social-engineering component of attacks such as rug pulls, which rely on manipulating user decisions rather than defeating the cryptographic layer. These are complementary areas of defense that remain necessary for a complete Web 3.0 security posture and are outside the scope of the authentication mechanism proposed here. Finally, as Section VI-E notes, full anonymity in the self-sovereign-identity sense also depends on the application-layer binding between public keys and real-world identity, which this protocol does not itself provide and which should be addressed jointly with a DID/verifiable-credential layer in any deployment.

XI. Conclusion

This study presented an authentication framework for decentralized Web 3.0 environments, built around Elliptic Curve Cryptography, an ephemeral Elliptic Curve ElGamal encoding scheme, and a mutual-authentication handshake that combines digital signatures over each party's long-term key with a two-ephemeral-key ECEG transport of the encoding parameters. The framework was evaluated against an explicit Dolev–Yao threat model with an explicit in-scope/out-of-scope adversary boundary (Table III); the informal security analysis indicates resistance to passive eavesdropping, replay, and, conditional on ephemeral-key erasure, later compromise of a long-term key, attributable to session-unique ephemeral parameters bound by the hardness of the Elliptic Curve Discrete Logarithm Problem. Resistance to man-in-the-middle substitution additionally requires the authenticated binding between long-term public keys and identities that Section VI-B supplies through a concrete decentralized-identifier/verifiable-credential mechanism. A protocol-level operation count and a computational complexity comparison against RSA, checked against the current NIST SP 800-57 equivalence table, showed the ECC security-to-key-size advantage while making clear that complete-protocol runtime, energy, and bandwidth performance remain to be established experimentally.

As Section II-D makes explicit, the ephemeral ECEG character-encoding step builds directly on prior work [2,3]; this study's own contribution is the protocol-level replacement of that prior work's static-key parameter transport with a signature-and-two-ephemeral-key-ECEG construction, together with the explicit Web 3.0-specific threat model, the DID/VC identity-binding mechanism, the requirements analysis, and the concrete formal-verification plan of Section VI-F built around it. Section IX further situated the framework's classical ECC foundation against the rapidly narrowing 2025–2026 quantum resource estimates for the elliptic-curve discrete logarithm problem — including a September 2026 physical-hardware estimate of roughly 19,397 qubits — and against current NIST post-quantum migration guidance, concluding that the framework remains an appropriate near- and medium-term choice but should be extended with a hybrid post-quantum key-establishment step ahead of the 2030–2035 deprecation horizon. Because the present evaluation is analytical, the five next steps identified in Section X — formal protocol verification, hardware-based benchmarking, anomaly-detection integration, hybrid post-quantum migration, and formal/empirical validation of the identity-binding mechanism — are necessary before the framework can be considered validated for production deployment. Subject to that validation, the proposed approach offers a practical building block for strengthening authentication and mitigating identity theft, phishing, and unauthorized access in self-sovereign, user-centric Web 3.0 identity ecosystems.

References

[1] Achary, R.; Raj, S.; Shelke, C.J. Integrated blockchain model for personally identifiable information system (IBPIIS): A secure approach. In Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 1–3 March 2024; pp. 1–7.

[2] Baccouri, S.; Farhat, H.; Gharbi, N.; Tahar, R.; Azzabi, T.; Attia, R. Ephemeral encoding message in the elliptic curve cryptography for IoT. In Proceedings of the 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, 22–25 March 2022; pp. 172–177.

[3] Baccouri, S.; Farhat, H.; Azzabi, T.; Attia, R. Lightweight authentication scheme based on Elliptic Curve El Gamal. *J. Inf. Telecommun.* 2024, *8*, 231–261.

[4] Dinarvand, N.; Barati, H. An efficient and secure RFID authentication protocol using elliptic curve cryptography. *Wirel. Netw.* 2019, *25*, 415–428.

[5] Lamrani Alaoui, H.; El Ghazi, A.; Zbakh, M.; Touhafi, A.; Braeken, A. A highly efficient ECC-based authentication protocol for RFID. *J. Sens.* 2021, *2021*, 8876766.

[6] Noori, D.; Shakeri, H.; Niazi Torshiz, M. An elliptic curve cryptosystem-based secure RFID mutual authentication for the Internet of things in healthcare environment. *EURASIP J. Wirel. Commun. Netw.* 2022, *2022*, 64.

[7] Sowjanya, K.; Dasgupta, M.; Ray, S. Elliptic curve cryptography based authentication scheme for internet of medical things. *J. Inf. Secur. Appl.* 2021, *58*, 102761.

[8] Khan, M.A.; Quasim, M.T.; Alghamdi, N.S.; Khan, M.Y. A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data. *IEEE Access* 2020, *8*, 52018–52027.

[9] Kumari, A.; Jangirala, S.; Abbasi, M.Y.; Kumar, V.; Alam, M. ESEAP: ECC-based secure and efficient mutual authentication protocol using smart card. *J. Inf. Secur. Appl.* 2019, *51*, 102443.

[10] Kumari, A.; Yahya Abbasi, M.; Kumar, V.; Khan, A.A. A secure user authentication protocol using elliptic curve cryptography. *J. Discrete Math. Sci. Cryptogr.* 2019, *22*, 521–530.

[11] Li, X.; Peng, J.; Niu, J.; Wu, F.; Liao, J.; Choo, K.R. A robust and energy efficient authentication protocol for industrial internet of things. *IEEE Internet Things J.* 2018, *5*, 1606–1615

[12] Panja, A.; Mondal, S.; Islam, K.A.; Ghosh, T.K.; Karforma, S. An efficient and secure fingerprint-based authentication scheme using elliptic curve cryptography. *Webology* 2022, *19*.

[13] Ramasamy, P.; Ranganathan, V.; Palanisamy, V.; Kadry, S. Securing one-time password generation using elliptic-curve cryptography with self-portrait photograph for mobile commerce application. *Multimed. Tools Appl.* 2020, *79*, 17081–17099.

[14] Sadri, M.J.; Asaar, M.R. An anonymous two-factor authentication protocol for IoT-based applications. *Comput. Netw.* 2021, *199*, 108460.

[15] Elias, E.P.; Santhanavijayan, A.; Janet, B.; Arul, K.R.J. OTP system based on ECC key exchange. In Proceedings of the 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 3–4 March 2022; pp. 1–6.

[16] Kaur, S.; Kaur, G.; Shabaz, M. A secure two-factor authentication framework in cloud computing. *Secur. Commun. Netw.* 2022, *2022*, 7540891.

[17] Islam, M.M.; Hossain, M.S.; Hasan, M.K.; Shahjalal, M.; Jang, Y.M. Design and implementation of high-performance ECC processors with unified point addition on twisted Edwards curve. *Sensors* 2020, *20*, 5148.

[18] Basu Roy, D.; Mukhopadhyay, D. High-speed implementation of ECC scalar multiplication in GF(p) for generic Montgomery curves. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 2019, *27*, 1587–1600.

[19] Lara-Nino, C.A.; Diaz-Perez, A.; Morales-Sandoval, M. Elliptic curve lightweight cryptography: A survey. *IEEE Access* 2018, *6*, 72514–72550.

[20] Prakash, M.; Ramesh, K. Blockchain-enabled decentralized lightweight authentication scheme with dynamic difficulty adjustment using ECC for IoT networks. *Cluster Comput.* 2025, *28*, 898.

[21] Liu, H.; Cao, J.; Cao, Y.; Bai, D.; Liang, J.; Li, R. EdenDID: an edge computing and blockchain-based decentralized identity system for Web3 applications and DePIN. *Cluster Comput.* 2025, *28*, 1048.

[22] Le, H.V.A.; Nguyen, Q.D.N.; Tadashi, N.; Tran, T.H. Blockchain-Based Decentralized Identity Management System with AI and Merkle Trees. *Computers* 2025, *14*, 289.

[23] Achary, R.; Shelke, C.J.; Marx, K.; Rajesh, A. Security implementation on IoT using CoAP and elliptical curve cryptography. *Procedia Comput. Sci.* 2023, *230*, 493–502.

[24] Zheng, J.; Wang, X.; Yang, Q.; Xiao, W.; Sun, Y.; Liang, W. A blockchain-based lightweight authentication and key agreement scheme for internet of vehicles. *Connect. Sci.* 2022, *34*, 1430–1453.

[25] Cremers, C.J.F. Scyther: Semantics and Verification of Security Protocols. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.

[26] Blanchet, B. Modeling and verifying security protocols with the applied pi calculus and ProVerif. *Found. Trends Priv. Secur.* 2016, *1*, 1–135.

[27] K. K. G. V.; Achary, R.; Shelke, C.J. Cloud data anomaly detection using machine learning. In Proceedings of the 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India, 18–19 April 2024; pp. 1245–1251.

[28] Kanjanapruthipong, W.; Boonkrong, S. Blockchain-Based Decentralised Authentication in Closed Environments. *Future Internet* 2025, *17*, 98.

[29] Goel, A.; Rahulamathavan, Y. A Comparative Survey of Centralised and Decentralised Identity Management Systems: Analysing Scalability, Security, and Feasibility. *Future Internet* 2025, *17*, 1.

[30] Ambati, S.; Adeel, K.; Myers, J.; Ivanov, N. SoK: Self-Sovereign Digital Identities. In Proceedings of the 31st ACM Symposium on Access Control Models and Technologies (SACMAT 2026), New York, NY, USA, 2026.

[31] National Institute of Standards and Technology. *Recommendation for Key Management: Part 1 General*; NIST Special Publication 800-57 Part 1 Revision 5; NIST: Gaithersburg, MD, USA, 2020.

[32] National Institute of Standards and Technology. *Transition to Post-Quantum Cryptography Standards*; NIST Internal Report 8547 (Initial Public Draft); NIST: Gaithersburg, MD, USA, 2024.

[33] Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* 1983, *29*, 198–208.

[34] Alotaibi, B. Cybersecurity Attacks and Detection Methods in Web 3.0 Technology: A Review. *Sensors* 2025, *25*, 342.

[35] Babbush, R.; Zalcman, A.; Gidney, C.; Broughton, M.; Khattar, T.; Neven, H.; Bergamaschi, T.; Drake, J.; Boneh, D. Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities: Resource Estimates and Mitigations. *PRX Quantum* 2026, *7*, 031001.

[36] Chevignard, C.; Fouque, P.-A.; Schrottenloher, A. Reducing the Number of Qubits in Quantum Discrete Logarithms on Elliptic Curves. In *Advances in Cryptology EUROCRYPT 2026*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2026; pp. 371–401.

[37] Luo, H.; Yang, Z.; Wang, Z.; Su, Y.; Li, T. Space-Efficient Quantum Algorithm for Elliptic Curve Discrete Logarithms with Resource Estimation. arXiv:2604.02311, 2026. Preprint.

[38] National Institute of Standards and Technology. *Announcing Issuance of Federal Information Processing Standards (FIPS) 203, 204, and 205*; Federal Register 2024-17956; NIST: Gaithersburg, MD, USA, 2024.

[39] Fiaz, F.; Muhammad, Z. QR-MetaSSI: A Quantum-Resistant Self-Sovereign Identity Framework for Metaverse Platforms. *J. Cybersecur. Priv.* 2026, *6*, 111.

[40] Radanliev, P.; Maple, C.; Santos, O. Complying with the NIST post-quantum cryptography standards and decentralizing artificial intelligence: methodology for quantum-resistant and privacy-preserving digital identity systems. *Front. Blockchain* 2025, *8*, 1702066.

[41] Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* 1987, *48*, 203–209.

[42] W3C. *Decentralized Identifiers (DIDs) v1.0*. W3C Recommendation, 19 July 2022.

[43] W3C. *Verifiable Credentials Data Model v2.0*. W3C Recommendation, 2025.

[44] Wuille, P.; Nick, J.; Ruffing, T. *BIP-340: Schnorr Signatures for secp256k1*. Bitcoin Improvement Proposals, 2020.

[45] Pornin, T. *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)*. IETF RFC 6979, 2013.

[46] Standards for Efficient Cryptography Group (SECG). *SEC 1: Elliptic Curve Cryptography, Version 2.0*. 2009.

[47] National Institute of Standards and Technology. *Recommendations for Discrete Logarithm-Based Cryptography: Elliptic Curve Domain Parameters*. NIST SP 800-186, 2023.

[48] Hammi, B.; Fayad, A.; Khatoun, R.; Zeadally, S.; Begriche, Y. A lightweight ECC-based authentication scheme for internet of things (IoT). *IEEE Syst. J.* 2020, *14*, 3440–3450.

[49] Li, M.; Hu, S. A Lightweight ECC-Based Authentication and Key Agreement Protocol for IoT with Dynamic Authentication Credentials. *Sensors* 2024, *24*, 7967.

[50] Rodriguez Garzon, S.; Natusch, D.; Philipp, A.; Küpper, A.; Einsiedler, H.J.; Schneider, D. DID Link: Authentication in TLS with Decentralized Identifiers and Verifiable Credentials. In Proceedings of the 2024 21st Annual International Conference on Privacy, Security and Trust (PST), 2024.

[51] Häner, T.; Tripier, F.; Young, J.; Naehrig, M.; Maksymov, A.; Alam, S.; Maslov, D.; Parrott, M.; de Sereville, Y.; Sullivan, J.; Webster, M.; Delfosse, N.; Gamble, J.; Roetteler, M. Computing 256-Bit Elliptic Curve Discrete Logarithms in 26 Days on a Fault-Tolerant Trapped-Ion Quantum Computer with 20,000 Qubits. arXiv:2609.05625, 2026. Preprint.

[52] Balfaqih, M.; Balfagih, Z.; Lytras, M.D.; Alfawaz, K.M.; Alshdadi, A.A.; Alsolami, E. A Blockchain-Enabled IoT Logistics System for Efficient Tracking and Management of High-Price Shipments: A Resilient, Scalable and Sustainable Approach to Smart Cities. *Sustainability* 2023, *15*, 13971.

[53] Balfaqih, M. Enhancing Security and Flexibility in Smart Locker Systems: A Multi-Authentication Approach with IoT Integration. In Proceedings of the 2024 21st Learning and Technology Conference (L&T), 2024; pp. 325–329.

[54] Tamarin Prover. *Tamarin Manual, Version 1.12*; Tamarin Prover Project, 2026.

Automated Generation of Arabic Procedural Court Orders in the Moroccan Legal Context

Saliha Yassine, Mohammed Essoujaa, Mustapha Esghir, Ouafaa Ibrihich
Faculty of Sciences, Mohammed V University in Rabat, Laboratory MIA-SI, Rabat, Morocco

***Abstract*—Automating the drafting of legal documents can reduce repetitive judicial workload and support procedural efficiency in e-Justice systems. However, Arabic legal text generation remains underexplored, particularly for Moroccan procedural court orders, which are characterized by highly structured language and recurring legal formulations. This study compares retrieval-based and transformer-based approaches for generating Moroccan procedural court orders using a corpus of 4,098 authentic orders collected from 2020 to 2025. Three approaches are evaluated: a TF-IDF-based Template Baseline, AraBART, and AraT5. A random stratified split and a stricter template-disjoint evaluation are used to assess both performance and generalization. Under the random split, the Template Baseline achieves the highest BLEU score (87.39), followed by AraBART (77.33) and AraT5 (51.88). In contrast, under the template-disjoint setting, AraBART outperforms the Template Baseline, achieving BLEU scores of 68.00 and 63.67, respectively, with a statistically significant difference ($p = 0.004$). These findings demonstrate that retrieval-based generation can benefit from repeated legal templates, whereas AraBART achieves better performance under the template-disjoint setting when exact templates are absent. The results highlight the importance of evaluation design and corpus structure when selecting generation approaches for standardized legal documents. A conceptual hybrid strategy combining retrieval, neural generation, and human validation is proposed for future e-Justice applications.**

Keywords*—Arabic legal NLP; legal text generation; moroccan procedural court orders; retrieval-based generation; AraBART; AraT5; e-Justice

I. Introduction

The rapid digitalization of judicial systems has increased interest in Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies for supporting judicial document drafting and administrative procedures. In Morocco, procedural court orders constitute an essential component of judicial administration, supporting routine legal procedures such as payment orders, notifications, deposits, and other administrative decisions. Unlike judicial judgments, these documents are generally concise, highly structured, and frequently follow standardized legal formulations. Automating their drafting has the potential to improve document consistency, accelerate routine judicial procedures, and support the development of intelligent e-Justice services while allowing legal professionals to focus on more complex judicial tasks [24].

Recent advances in transformer-based models and large language models have significantly improved automatic text generation across numerous domains, including legal NLP [1]. Arabic generative models such as AraBART and AraT5 have demonstrated promising performance by leveraging large-scale pretraining and contextual language representations. However, applying these models to procedural legal documents presents a unique challenge. Because many procedural court orders contain recurring legal expressions and standardized templates, it remains unclear whether computationally intensive neural generation models consistently provide advantages over simpler retrieval-based approaches. Consequently, selecting the most appropriate generation paradigm depends not only on the model architecture but also on the intrinsic characteristics of the legal corpus.

Existing research on legal text generation has primarily focused on judgment summarization, legal question answering, statute retrieval, and general legal document drafting, with most studies conducted for English- or Chinese-language legal systems. Arabic legal NLP remains comparatively underexplored, particularly for generative tasks involving authentic judicial documents. Although recent Arabic transformer models, including AraBART and AraT5, have achieved encouraging results across several natural language generation tasks, their effectiveness for generating procedural court orders has received little attention. Furthermore, previous studies rarely examine how corpus repetitiveness influences the comparative performance of retrieval-based and neural generation methods in highly standardized legal domains.

This observation reveals an important research gap. To the best of our knowledge, no prior study has reported a systematic comparison of retrieval-based and transformer-based generation for Moroccan procedural court orders while explicitly analyzing the influence of corpus repetition on model selection. Addressing this question is particularly important because highly repetitive legal corpora may favor retrieval-based methods when relevant templates are available in the retrieval database, whereas neural generation models may achieve better performance when exact decision templates are absent from the training data. Understanding this trade-off is essential for designing efficient, reliable, and practically deployable legal text generation systems for Moroccan courts and other legal environments characterized by standardized procedural documents.

To address this research gap, this study presents a comprehensive comparative study of retrieval-based and transformer-based approaches for generating Moroccan procedural court orders. Rather than assuming that increasingly complex neural models always produce superior results, this work investigates how corpus characteristics, particularly the degree of textual repetition, influence the effectiveness of different generation paradigms.

The main contributions of this study are summarized as follows:

- Use and systematic characterization of a corpus comprising 4,098 authentic Moroccan procedural court orders from 2020 to 2025, representing a systematically described dataset for Arabic legal text generation in this domain.
- Comprehensive comparative evaluation of a TF-IDF-based Template Baseline, AraBART, and AraT5 using complementary automatic evaluation metrics, enabling a systematic assessment of retrieval-based and transformer-based generation methods.
- Corpus repetition analysis demonstrating how the degree of textual repetition affects the relative performance of retrieval-based and transformer-based generation across different evaluation settings.
- A conceptual hybrid generation strategy derived from the experimental findings, recommending retrieval-based generation for highly repetitive cases and transformer models for more diverse drafting scenarios under human supervision, providing guidance for future intelligent e-Justice systems.

The remainder of this study is organized as follows. Section II reviews related work on Arabic legal NLP and legal text generation. Section III describes the proposed methodology, including corpus construction, preprocessing, generation models, and evaluation metrics. Section IV presents and discusses the experimental results. Finally, Section V concludes the study and outlines directions for future research.

II. Related Work

Legal Natural Language Processing (Legal NLP) has rapidly evolved from rule-based systems to transformer-based architectures, enabling significant advances in legal document classification, information extraction, judgment prediction, and text generation. Among these tasks, legal text generation has recently attracted increasing attention because of its potential to support judicial drafting, reduce repetitive workload, and improve the efficiency of e-Justice systems.

Domain-specific pre-training has proven effective across various legal NLP applications. LEGAL-BERT [2], which builds upon the BERT architecture [20], achieves strong performance for European court judgment prediction, while Zhong et al. [3] examined the application of NLP methods to several tasks in the Chinese legal system. Large-scale legal datasets such as CAIL2018 [16] have further supported research on judgment prediction. In legal text generation, Feijo et al. [4] demonstrated the feasibility of GPT-2 for automatic contract generation in Brazilian law. In the Arabic NLP domain, AraBERT [5] established a strong baseline for Arabic language understanding, whereas AraBART [6] and AraT5 [7] introduced transformer-based sequence-to-sequence architectures that have shown promising performance in Arabic text generation and summarization tasks. AraBART is grounded in the BART architecture [17] and AraT5 in the T5 text-to-text framework [18], both building on the foundational Transformer architecture [21]. Bidirectional Arabic pretraining models such as ARBERT and MARBERT [22] and comprehensive Arabic NLP resources [25] have further supported the development of these models.

Retrieval-based and case-based approaches have also been explored as alternatives to fully generative systems, particularly in domains where documents exhibit strong structural regularities. Such approaches can preserve established domain-specific formulations by reusing previously validated outputs. However, their effectiveness for Moroccan procedural court orders in Arabic has not been systematically compared with neural generation models.

More recently, large language models and retrieval-enhanced generation have further expanded the potential of legal text generation. General-purpose LLMs such as GPT-3 [23] and Llama 2 [8] and retrieval-augmented generation frameworks [9] illustrate the growing trend toward combining knowledge retrieval with neural generation. Dedicated benchmarks such as LegalBench-RAG [10] and Arabic-specific models such as AceGPT [11] further demonstrate the breadth of this research landscape. Although these approaches are promising, their application to highly standardized Arabic procedural court orders remains largely unexplored. Table I summarizes representative studies in Legal NLP and Arabic text generation. Existing research has mainly focused on legal judgment prediction, legal document classification, contract generation, and general-purpose Arabic text generation. However, the automatic generation of Moroccan procedural court orders has not yet been systematically investigated. Moreover, previous studies have not examined how corpus repetitiveness influences the relative effectiveness of retrieval-based and transformer-based generation approaches. To address this gap, the present work investigates how corpus repetitiveness influences the comparative performance of retrieval-based and transformer-based generation, providing practical guidance for designing efficient Arabic legal text generation systems for highly structured legal domains.

TABLE I. Representative Studies in Legal NLP and Arabic Text Generation

Study	Domain	Task	Main Method
Chalkidis et al. [2]	EU courts	Judgment prediction	LEGAL-BERT
Zhong et al. [3]	Chinese legal system	Multiple legal NLP tasks	Legal NLP methods
Feijo et al. [4]	Brazilian law	Contract generation	GPT-2
Antoun et al. [5]	Arabic NLP	Text classification	AraBERT
Kamal et al. [6]	Arabic NLP	Text generation	AraBART
Nagoudi et al. [7]	Arabic NLP	Text generation	AraT5
Sulea et al. [12]	French law	Legal classification	SVM + TF-IDF
Medvedeva et al. [13]	ECHR	Decision prediction	Linear SVM + textual n-grams

Although transformer-based models have demonstrated strong performance in Arabic text generation, their application to highly repetitive procedural legal documents raises a methodological question: whether generative flexibility provides a genuine advantage when valid judicial formulations

are already available in historical records. Retrieval-based approaches preserve established legal formulations and can reduce the risk of generative deviations at low computational cost, but they are limited in their ability to generate formulations unseen in the training corpus. Transformer-based approaches offer greater generative flexibility and can adapt to novel inputs, but remain exposed to hallucination and semantic deviation. This trade-off motivates the use of TF-IDF character n-gram retrieval, rather than a purely generative approach, as a baseline for highly standardized procedural court orders.

III. METHOD

A. Dataset

We compiled a corpus of 4,098 authentic Moroccan procedural court orders issued by first-instance courts between 2020 and 2025. The corpus was previously used for a classification study of Moroccan procedural court orders [26]; the present work addresses a distinct research question focused on automatic text generation and the comparative analysis of retrieval-based and transformer-based approaches. Each document consists of two main sections: faits (procedural facts describing the legal request) and décision (the judicial ruling issued by the court). In this study, the faits section serves as the input to all generation models, while the corresponding décision is used as the reference output during training and evaluation. Table II summarizes the main characteristics of the corpus. The corpus covers four procedural categories representing the most frequent types of procedural court orders handled by Moroccan first-instance courts.

TABLE II. CORPUS STATISTICS

Statistic	Value
Total documents	4,098
Training set	3,278 (80%)
Validation set	410 (10%)
Test set	410 (10%)
Average facts length (characters)	323
Average decision length (characters)	159
Number of procedural categories	4
Global repetition rate (training set)	64.0%
Random seed	42

To ensure a fair, reproducible, and representative evaluation, the corpus was partitioned using a stratified 80/10/10 split, resulting in 3,278 training documents, 410 validation documents, and 410 test documents. Stratification preserves the distribution of procedural categories across all subsets, while a fixed random seed (42) ensures the reproducibility of the data partition. Because the corpus contains repeated judicial formulations, the random stratified split may place identical decision formulations in both the training and test sets. This setting reflects the intended retrieval scenario but may favor retrieval-based methods on highly standardized cases. To address this limitation, an additional template-disjoint evaluation was conducted. Decision texts were grouped by their exact string content after normalizing whitespace, and a group-based split (scikit-learn's GroupShuffleSplit) was used to ensure that all documents sharing an identical decision formulation were assigned entirely to the same subset (training, validation, or test), thereby preventing any decision string from appearing in more than one split.

This evaluation provides a more stringent assessment of generalization beyond previously observed judicial templates. The resulting template-disjoint partition contained 3,364 training documents, 354 validation documents, and 380 test documents.

B. Corpus Analysis: Repetition Structure

A preliminary analysis of the training corpus reveals a global repetition rate of 64.0%, corresponding to 1,180 unique judicial decisions among 3,278 training documents. In this study, a repeated decision refers to an identical judicial ruling appearing in multiple procedural court orders after whitespace normalization. The repetition rate was computed by comparing unique decision texts within each procedural category, providing a quantitative measure of the degree of standardization in judicial drafting.

Fig. 1 illustrates the distribution of procedural categories in the training set, while Table III summarizes the corresponding repetition rates, showing that rejection orders exhibit the highest repetition (89.9%), followed by validation-of-warning orders (75.9%). In contrast, payment orders show considerably greater textual diversity, with a repetition rate of only 27.0%.

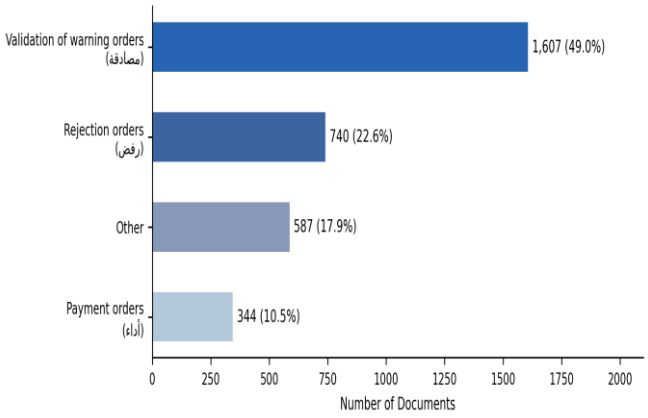


Fig. 1. Distribution of procedural court order categories in the training set (n = 3,278).

TABLE III. REPETITION ANALYSIS OF THE TRAINING CORPUS BY PROCEDURAL CATEGORY

Category	Documents	Unique Decisions	Repetition Rate (%)
Validation of warning orders (مصادقة)	1,607	388	75.9%
Rejection orders (رفض)	740	75	89.9%
Payment orders (أداء)	344	251	27.0%
Other	587	466	20.6%
Total (training set)	3,278	1,180	64.0%

These observations indicate that corpus repetition is a key factor in selecting an appropriate generation paradigm and

motivate the inclusion of a retrieval-based Template Baseline in the comparative evaluation.

Rejection orders exhibit the highest repetition rate (89.9%), reflecting the highly formulaic nature of procedural refusals under Moroccan procedural law. Validation-of-warning orders (75.9%) follow a similarly standardized structure. This observation explains the strong performance of the Template Baseline.

C. Systems

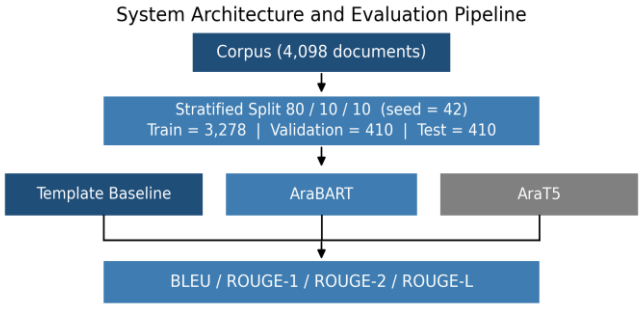


Fig. 2. Experimental framework for retrieval-based and transformer-based generation of Moroccan procedural court orders.

Fig. 2 illustrates the main experimental framework used for the random-split evaluation. An additional template-disjoint evaluation was conducted as described in Section III-A. Three complementary approaches were evaluated: a retrieval-based Template Baseline, AraBART, and AraT5. The retrieval-based system represents a non-neural approach that exploits the high repetition rate of procedural court orders, whereas AraBART and AraT5 represent neural sequence-to-sequence architectures capable of generating more flexible judicial texts.

1) Template baseline: For each test document, the most similar training document is retrieved using TF-IDF with character n-grams (3–5) and cosine similarity. The retrieved document's judicial decision is then returned as the generated output. Character n-grams were selected because they are more robust than word-based representations for Arabic text, effectively handling morphological variations, affixes, spelling differences, and tokenization inconsistencies while preserving lexical similarity between highly repetitive legal documents. This approach requires no neural training and directly exploits the standardized nature of procedural court orders.

2) AraBART: We fine-tuned moussaKam/AraBART [6], a BART-based sequence-to-sequence model [17] pre-trained on approximately 70 GB of Arabic text. AraBART was selected because it provides an Arabic sequence-to-sequence architecture designed for text generation and has demonstrated strong performance on Arabic generation tasks. The faits section is provided as input, while the corresponding décision is generated as output, with both input and output sequences limited to 128 tokens.

3) AraT5: We fine-tuned UBC-NLP/AraT5-base-title-generation [7], an Arabic adaptation of the T5 text-to-text framework [18]. AraT5 was included as a complementary Arabic text-to-text baseline for comparison with AraBART. Although this checkpoint was originally developed for title-generation tasks, it was included here as a complementary Arabic T5 baseline to assess its transferability to procedural legal text generation. The same input-output configuration and maximum sequence length were used to ensure a fair comparison with AraBART. A preliminary experiment with mT5-small [19] did not produce coherent Arabic judicial text and was therefore excluded from the main comparison; this result should not be interpreted as a general limitation of the mT5 architecture.

D. Hyperparameters

The hyperparameters used for fine-tuning AraBART and AraT5 are summarized in Table IV. To ensure a fair comparison between both transformer models, identical training settings were adopted whenever applicable while maintaining computational feasibility on the available hardware.

TABLE IV. TRAINING HYPERPARAMETERS

Parameter	AraBART	AraT5
Pre-trained model	moussaKam/AraBART	UBC-NLP/AraT5-base-title-generation
Training epochs	3	3
Batch size	8	4
Max input/output length (tokens)	128 / 128	128 / 128
Learning rate	5×10^{-5}	5×10^{-5}
Warmup steps	100	100
Precision	fp16	fp16
Hardware	Kaggle T4 GPU	Kaggle T4 GPU
Optimizer	AdamW	AdamW

E. Evaluation Metrics

The generated judicial decisions were evaluated using the BLEU [14] and ROUGE [15] metrics, which are widely adopted for automatic text generation evaluation. BLEU measures modified n-gram precision with a brevity penalty by comparing the generated text with the reference decision, whereas ROUGE evaluates recall-based lexical overlap. Specifically, ROUGE-1, ROUGE-2, and ROUGE-L were computed to assess unigram, bigram, and longest common subsequence similarity, respectively.

Because retrieval-based and neural generation models exhibit different generation behaviors, multiple evaluation metrics were employed to provide a comprehensive assessment. BLEU is particularly informative for highly repetitive legal documents where exact lexical matching is expected, while ROUGE captures partial lexical overlap when generated texts differ from the reference, complementing BLEU in cases where exact n-gram matching is limited.

Because Arabic is morphologically rich, BLEU and ROUGE may underestimate semantic similarity when equivalent legal expressions differ only in their morphological or orthographic forms. Therefore, these metrics are interpreted jointly with the qualitative analysis presented in Section IV-C to provide a more comprehensive evaluation of generation quality.

IV. Results and Discussion

A. Quantitative Comparison of Generation Models

TABLE V. Performance Comparison on 410 Test Documents

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Template Baseline	**87.39**	**21.30**	**15.29**	**21.14**
AraBART	**77.33**	**20.01**	**13.92**	**20.01**
AraT5	**51.88**	**19.00**	**12.81**	**19.00**

Table V summarizes the quantitative performance of the three evaluated generation approaches on the 410-document test set. The retrieval-based Template Baseline achieves the highest overall performance, obtaining a BLEU score of 87.39, followed by AraBART (77.33) and AraT5 (51.88). Similar performance trends are observed across the ROUGE metrics, where the Template Baseline consistently outperforms both transformer-based models.

Statistical significance analysis: To assess whether the observed differences between the generation approaches were statistically significant, a paired bootstrap analysis was conducted over the 410 test documents using 2,000 bootstrap replications. The analysis was performed on document-level BLEU scores, with each comparison using the same test documents for both methods. The Template Baseline significantly outperformed AraBART, with a mean BLEU difference of 10.07 points (95% CI: [7.74, 12.34], $p < 0.001$). Similarly, the Template Baseline significantly outperformed AraT5 by 35.51 BLEU points (95% CI: [32.59, 38.38], $p < 0.001$). AraBART also significantly outperformed AraT5, with a difference of 25.45 BLEU points (95% CI: [23.03, 27.66], $p < 0.001$). These results confirm that the observed performance differences are statistically significant under the present test-set configuration.

The results under the random split initially show a strong advantage for the Template Baseline, which is consistent with the highly repetitive nature of Moroccan procedural court orders. However, the template-disjoint evaluation provides a more stringent test of generalization and reverses this ranking, with AraBART achieving a significantly higher BLEU score than the Template Baseline. This contrast indicates that the apparent superiority of retrieval-based generation under the random split is partly associated with the availability of repeated decision formulations across the training and test sets. Since many judicial decisions follow standardized legal formulations, retrieving the most similar precedent frequently produces an output that closely matches the reference decision. Consequently, retrieval-based generation can be particularly effective for highly standardized procedural court orders when sufficiently similar validated precedents are available in the retrieval corpus.

Among the evaluated checkpoints, AraBART achieves higher BLEU, ROUGE-1, and ROUGE-L scores than the selected AraT5 checkpoint (UBC-NLP/AraT5-base-title-generation), while AraT5 obtains a slightly higher ROUGE-2 score than AraBART under the template-disjoint setting; this discrepancy may partly reflect checkpoint mismatch rather than a general architectural disadvantage of AraT5.

Under the random split, the Template Baseline achieves the highest scores, while the relatively small differences in ROUGE-1 (21.30 vs. 20.01) and ROUGE-L (21.14 vs. 20.01) indicate that transformer models may generate expressions that differ lexically from the reference text. This observation highlights the complementary roles of BLEU and ROUGE in evaluating legal text generation, particularly when comparing retrieval-based and neural generation approaches.

B. Template-Disjoint Evaluation

To assess the generalization of the evaluated approaches beyond repeated decision templates, an additional template-disjoint evaluation was conducted. In contrast to the initial random stratified split, the template-disjoint protocol prevents documents sharing an identical decision string from being assigned to different subsets. The resulting test set contains 380 documents and provides a more stringent assessment of model generalization to decision formulations not represented in the training set. The AraBART and AraT5 models were re-trained using the corresponding template-disjoint training set, while the Template Baseline retrieved candidate decisions only from that same training set.

The results reveal a substantially different performance pattern from that obtained under the random split. As shown in Table VI, AraBART achieved the highest BLEU score (68.00), followed by the Template Baseline (63.67) and AraT5 (39.89). AraBART also obtained higher ROUGE-1, ROUGE-2, and ROUGE-L scores than the Template Baseline, indicating better lexical overlap with the reference decisions under the template-disjoint setting.

TABLE VI. Template-Disjoint Evaluation on 380 Test Documents

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Template Baseline	**63.67**	**19.37**	**8.99**	**19.20**
AraBART	**68.00**	**26.32**	**16.54**	**26.32**
AraT5	**39.89**	**25.61**	**17.09**	**25.61**

The difference between the two evaluation settings is particularly informative. Under the random split, the Template Baseline achieved a BLEU score of 87.39 compared with 77.33 for AraBART. Under the template-disjoint split, however, AraBART outperformed the Template Baseline by 4.33 BLEU points. This contrast indicates that retrieval-based generation benefits substantially from the presence of repeated decision formulations across the training and test sets, whereas AraBART achieves better performance when exact decision templates are excluded from the training data.

To determine whether this difference was statistically significant, a paired bootstrap analysis with 2,000 replications was performed on document-level BLEU scores using the same 380 test documents for each method. AraBART significantly outperformed the Template Baseline, with a BLEU difference of 4.33 points (95% CI: [1.60, 6.95], $p = 0.004$). The Template Baseline also significantly outperformed AraT5 by 23.78 BLEU

points ($p < 0.001$), while AraBART significantly outperformed AraT5 by 28.11 BLEU points ($p < 0.001$). These results confirm that the advantage of AraBART over the retrieval-based baseline under the template-disjoint setting is statistically significant.

Overall, the template-disjoint evaluation provides complementary evidence to the random-split results. Rather than indicating that one generation paradigm is universally superior, the two-evaluation settings show that the relative performance of retrieval-based and generative approaches is strongly influenced by the degree of template overlap between training and test data.

C. Qualitative Analysis of Generated Judicial Decisions

TABLE VII. QUALITATIVE EXAMPLES — RANDOM-SPLIT TEST SET (EXCERPTS)

Facts (excerpt)	Reference	Template	AraBART	Observation
طلب مقدم بتاريخ 06 يناير 2022 يلتمس تبليغ إنذار بأداء...	نوافق على الطلب و نأمر بتبليغ الإنذار المرفق بالمقال المطلوبة. We approve the request and order the notification of the warning attached to the petition to the respondent (fem.).	نوافق على الطلب و نأمر بتبليغ الإنذار المرفق بالمقال للمطلوب. We approve the request and order the notification of the warning attached to the petition to the respondent (masc.).	نوافق على الطلب و نأمر بتبليغ الإنذار بالأداء المرفق بالمقال. We approve the request and order the notification of the payment warning attached to the petition.	Minor morphological variation
مقال مسجل بتاريخ 07 نونبر 2023 يلتمس الإذن بتبليغ إنذار بالأداء...	نوافق على الطلب ونأمر بتبليغ الإنذار بالأداء المرفق بالمقال للمطلوب. We approve the request and order the notification of the payment warning attached to the petition to the respondent.	نوافق على الطلب ونأمر بتبليغ الإنذار بالأداء المرفق بالمقال للمطلوب. [Identical to Reference]	نوافق على الطلب ونقرر المصادقة على الإنذار المرفق بالطلب. We approve the request and decide to validate the warning attached to the application.	Semantic deviation (judicial action different)
مقال مقدم بتاريخ 28 نونبر 2023 يلتمس تبليغ إنذار بأداء دين...	نوافق على الطلب و نأمر بتبليغ الإنذار بالأداء المرفق بالمقال للمطلوب. We approve the request and order the notification of the payment warning attached to the petition to the respondent.	نوافق على الطلب و نأمر بتبليغ الإنذار بالأداء المرفق بالمقال للمطلوب. [Identical to Reference]	نوافق على الطلب و نأمر بتبليغ الإنذار المرفق بالمقال للجهة المطلوبة. We approve the request and order the notification of the warning attached to the petition to the respondent party.	Alternative wording, preserved legal meaning

Table VII focuses on the Template Baseline and AraBART, the two approaches selected for qualitative comparison, to illustrate the contrast between retrieval-based and neural generation. AraT5 is not included because it obtained lower BLEU, ROUGE-1, and ROUGE-L scores than AraBART across the two evaluation settings, although it slightly outperformed AraBART in ROUGE-2 under the template-disjoint setting. Table VII presents representative examples illustrating the qualitative behavior of the retrieval-based Template Baseline and AraBART on held-out test documents. While automatic metrics quantify overall performance, these examples illustrate the different behaviors of retrieval-based and transformer-based generation.

The Template Baseline frequently retrieves judicial decisions that are identical or nearly identical to the reference text, reflecting the high level of standardization observed in Moroccan procedural court orders. In the second and third examples, the retrieved decisions exactly match the reference output. In the first example, only a minor morphological agreement variation, المطلوب instead of المطلوبة, is observed, while the underlying procedural action remains unchanged.

In contrast, AraBART generally generates fluent and grammatically well-formed judicial text, but it may introduce semantic deviations. As illustrated in the first and third examples, the generated decisions preserve the overall legal intent while using alternative legal expressions. However, the second example shows that AraBART may also generate a different judicial action (validation of the warning) instead of the expected notification of the warning, leading to a legally inappropriate output despite its linguistic fluency.

Overall, the qualitative analysis complements the quantitative results reported in Table V and illustrates the behavior of the two approaches under the random-split setting. Retrieval-based generation can produce highly faithful outputs when sufficiently similar precedents are available, whereas transformer-based generation provides greater linguistic flexibility but may remain susceptible to semantic deviations in legally sensitive contexts.

To complement the illustrative examples in Table VII, all 410 test documents were automatically classified into seven output categories by comparing each prediction with its reference decision, using text similarity, judicial-action keywords, and monetary-amount comparison. This automatic, similarity-based classification is not equivalent to expert legal review, but it provides a systematic, corpus-wide complement to the qualitative examples above. Table VIII summarizes the results for AraBART and the Template Baseline.

TABLE VIII. SYSTEMATIC ERROR ANALYSIS ACROSS THE RANDOM-SPLIT TEST SET

Output Category	AraBART	Template	Description
Exact match	30.5%	60.7%	Output identical to the reference
Lexical variation	26.1%	8.8%	Alternative phrasing, meaning preserved
Morphological variation	23.4%	10.7%	Minor difference (agreement, punctuation)
Numerical substitution	9.8%	13.2%	Amount (and sometimes identity) differs from reference
Procedural omission	2.4%	1.2%	Expected amount or element missing
Semantic deviation	0.2%	0.5%	Different judicial action (e.g., validation instead of notification)
Other/major deviation	7.6%	4.9%	Differences not classified above

A closer analysis of AraBART's outputs further reveals that 9.8% of test documents were automatically categorized as involving numerical or beneficiary-identity substitution. In several cases, recurring names or amounts were observed across different inputs, suggesting that the model may sometimes rely on frequently observed patterns rather than fully conditioning its output on the input facts. This observation requires further expert validation and should therefore be interpreted as an indication rather than a definitive causal explanation. Although automatic metrics such as BLEU and ROUGE do not necessarily distinguish this type of error from a minor lexical variation, it may constitute a materially more serious risk in a judicial deployment context, reinforcing the need for systematic human validation before any generated decision is adopted in practice.

D. Practical Implications and Recommended Generation Strategy

The combined experimental results demonstrate that no single generation approach is universally optimal for Moroccan procedural court orders; performance depends substantially on the availability of repeated templates and the evaluation setting.

For highly standardized procedural categories, where sufficiently similar and validated precedents are available, retrieval-based generation can be particularly suitable. By retrieving an existing precedent, the Template Baseline preserves the original judicial wording and can achieve strong quantitative performance without requiring computationally intensive neural inference. However, the template-disjoint evaluation shows that this advantage does not necessarily extend to cases where exact decision templates are absent from the training data.

For procedural categories exhibiting greater textual diversity, transformer-based models offer increased linguistic flexibility by generating decisions from the procedural facts; however, their outputs require legal validation, as semantic deviations may occur. Among the evaluated transformer models, AraBART achieves the highest BLEU, ROUGE-1, and ROUGE-L scores in the template-disjoint evaluation, while AraT5 obtains a slightly higher ROUGE-2 score than AraBART, making AraBART the preferred neural alternative when retrieval-based generation is less appropriate.

Based on the complementary results obtained under the random and template-disjoint evaluation settings, we propose a conceptual hybrid generation strategy for future practical e-Justice applications, illustrated in Fig. 3. Retrieval-based generation should be employed when a sufficiently similar precedent is identified with high confidence. In contrast, a transformer model such as AraBART may be preferable when exact templates are unavailable. In all cases involving low-confidence retrievals or legally sensitive decisions, human validation remains essential before the generated text is adopted in judicial practice.

The present study is limited to Moroccan procedural court orders and therefore does not establish cross-jurisdictional generalizability. Differences in legal terminology, procedural rules, document structure, and drafting conventions may substantially affect model performance in other jurisdictions. Future studies should evaluate the proposed methodology on procedural legal corpora from other Arabic-speaking jurisdictions. Due to the legal and privacy constraints associated with judicial documents, the full corpus cannot be publicly released.

A further limitation concerns the absence of systematic human evaluation by legal professionals. The qualitative analysis presented in Table VII and the automatic error categorization in Table VIII provide complementary evidence, but they cannot replace expert assessment of legal correctness, semantic fidelity, and procedural appropriateness. A future study should involve legal experts and report inter-annotator agreement to provide a more rigorous human-centered evaluation.

The study also does not provide a systematic quantitative analysis of computational cost, including training time, inference latency, and memory consumption. Such measurements should be incorporated in future experiments to provide a more complete assessment of deployment feasibility.

Although the present study focuses on Moroccan procedural court orders, the proposed strategy may also be applicable to other legal domains characterized by varying levels of textual standardization. Future work may investigate adaptive systems capable of automatically selecting the most appropriate generation paradigm according to corpus characteristics and retrieval confidence.

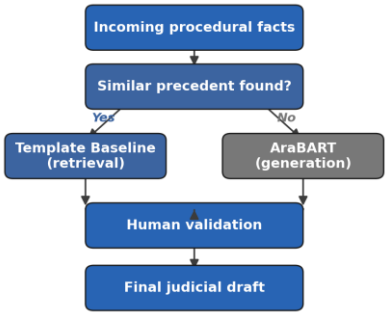


Fig. 3. Conceptual hybrid generation strategy recommended for Moroccan procedural court orders.

V. Conclusion

This study presented a comparative study of retrieval-based and transformer-based generation for Moroccan procedural court orders. Using a corpus of 4,098 authentic Moroccan procedural court orders, we compared a retrieval-based Template Baseline with two Arabic sequence-to-sequence models, AraBART and AraT5, under complementary evaluation settings.

Under the random stratified split, the Template Baseline achieved the highest BLEU score (87.39), reflecting the strong repetition present in the corpus. However, the additional template-disjoint evaluation produced a different result: AraBART achieved a BLEU score of 68.00 compared with 63.67 for the Template Baseline, and the 4.33-point difference was statistically significant ($p = 0.004$). This contrast demonstrates that retrieval-based performance is strongly influenced by the presence of repeated templates across training

and test data, whereas AraBART achieves better performance under the template-disjoint setting.

These findings show that no single generation paradigm is universally optimal. Retrieval-based generation is advantageous when reliable and sufficiently similar precedents are available, while transformer-based generation may be advantageous when the required formulations are not directly represented in the retrieval corpus. The evaluation protocol itself therefore plays an important role in determining the apparent effectiveness of each approach.

Beyond the quantitative comparison, this study shows that the effectiveness of a generation approach depends on the degree of textual repetition within the target corpus. Retrieval-based generation is particularly suitable when sufficiently similar and validated precedents are available, whereas transformer-based models provide greater flexibility and may be advantageous when exact legal formulations are not represented in the retrieval corpus. Based on these findings, we propose a conceptual hybrid generation strategy, intended as a practical guideline that combines retrieval-based generation, neural text generation, and human validation to support future e-Justice applications; this strategy has not yet been implemented or empirically evaluated as an integrated system.

Future work will investigate retrieval-confidence estimation, automatic selection between retrieval-based and neural generation, and human-in-the-loop validation. Recent developments in legal retrieval-augmented generation [10] and Arabic large language models such as AceGPT [11] also provide promising directions for extending the proposed strategy. The proposed strategy will additionally be evaluated on other categories of Moroccan legal documents to assess its generalizability.

VI. Declaration on Generative AI

Generative AI tools were used for language refinement and editorial assistance during manuscript preparation. The authors critically reviewed and validated the resulting text and remain fully responsible for the scientific content, methodology, results, and conclusions of this work.

Acknowledgment

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA), and the CNRST of Morocco, grant number Alkhawarizmi/2020/25.

References

[1] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, and M. J. Bommarito, “Natural Language Processing in the Legal Domain,” arXiv preprint arXiv:2302.12039, 2023.

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets Straight Out of Law School," in Findings of EMNLP, Online, Nov. 2020, pp. 2898–2904, doi: 10.18653/v1/2020.findings-emnlp.261.

[3] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "How Does NLP Benefit Legal System," in Proc. 58th Annu. Meeting ACL, Online, Jul. 2020, pp. 5218–5230, doi: 10.18653/v1/2020.acl-main.466.

[4] D. Feijo and V. P. Moreira, "Automatic Contract Generation with GPT-2," in Proc. Brazilian Conf. Intell. Syst. (BRACIS), Nov. 2021, pp. 532–546, doi: 10.1007/978-3-030-91699-2_36.

[5] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," in Proc. 4th Workshop Open-Source Arabic Corpora (OSACT4), Marseille, France, May 2020, pp. 9–15.

[6] I. Kamal and M. Abdul-Mageed, "AraBART: A Pretrained Arabic Sequence-to-Sequence Model for Abstractive Summarization," in Proc. 6th Arabic Natural Language Processing Workshop, Punta Cana, Nov. 2021, pp. 231–236.

[7] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "AraT5: Text-to-Text Transformers for Arabic Language Generation," in Proc. 60th Annu. Meeting ACL, Dublin, Ireland, May 2022, pp. 628–647, doi: 10.18653/v1/2022.acl-long.47.

[8] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” arXiv preprint arXiv:2307.09288, 2023.

[9] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[10] N. Pipitone and G. Houir Alami, “LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain,” arXiv preprint arXiv:2408.10343, 2024.

[11] H. Huang et al., “AceGPT, Localizing Large Language Models in Arabic,” in Proc. 2024 Conf. North Am. Chapter ACL (NAACL-HLT), Mexico City, Mexico, Jun. 2024, pp. 8139–8163, doi: 10.18653/v1/2024.naacl-long.450.

[12] O. Sulea, M. Zampieri, M. Vela, and J. van Genabith, "Exploring the Use of Text Classification in the Legal Domain," in Proc. Workshop Automated Semantic Analysis Information Law (ASAIL), London, UK, Jun. 2017, pp. 1–9.

[13] M. Medvedeva, M. Vols, and M. Wieling, "Using Machine Learning to Predict Decisions of the European Court of Human Rights," Artif. Intell. Law, vol. 28, no. 2, pp. 237–266, Jun. 2020, doi: 10.1007/s10506-019-09255-y.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in Proc. 40th Annu. Meeting ACL, Philadelphia, PA, USA, Jul. 2002, pp. 311–318, doi: 10.3115/1073083.1073135.

[15] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Proc. Workshop Text Summarization Branches Out, Barcelona, Spain, Jul. 2004, pp. 74–81.

[16] C. Xiao et al., "CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction," arXiv:1807.02478 [cs.AI], Jul. 2018.

[17] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proc. 58th Annu. Meeting ACL, Online, Jul. 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[18] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.

[19] L. Xue et al., "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer," in Proc. 2021 Conf. North Am. Chapter ACL (NAACL-HLT), Online, Jun. 2021, pp. 483–498, doi: 10.18653/v1/2021.naacl-main.41.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[21] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[22] M. Abdul-Mageed et al., "ARBERT and MARBERT: Deep Bidirectional Transformers for Arabic," in Proc. 59th Annu. Meeting ACL, Online, Aug. 2021, pp. 7088–7105, doi: 10.18653/v1/2021.acl-long.551.

[23] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, Online, Dec. 2020, pp. 1877–1901.

[24] K. D. Ashley, Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age. Cambridge, UK: Cambridge Univ. Press, 2017.

[25] N. Habash, Introduction to Arabic Natural Language Processing. San Rafael, CA, USA: Morgan & Claypool Publishers, 2010, doi: 10.2200/S00277ED1V01Y201008HLT010.

[26] S. Yassine, O. Ibrihich, M. Essoujaa, and M. Esghir, "Automated Classification of Moroccan Procedural Court Orders Using CAMeL-BERT," International Journal of Advanced Computer Science and Applications, vol. 17, no. 7, 2026, doi: 10.14569/IJACSA.2026.0170719.

Semi-Supervised Rice Leaf Disease and Pest Classification Using IoT Field Images

Abdul Azis[1]*, Abdul Fadlil[2], Tole Sutikno[3]
Informatics Study Program, Ahmad Dahlan University, Yogyakarta, Indonesia[1]
Information System Department, Amikom Purwokerto University, Purwokerto, Indonesia[1]
Information System Study Program, Amikom Purwokerto University, Purwokerto, Indonesia[2]
Department of Electrical Engineering, Ahmad Dahlan University, Yogyakarta, Indonesia[3]

***Abstract*—Rice is a vital crop for global food security, but its productivity is frequently threatened by diseases and pests, necessitating rapid and accurate detection. This study presents a semi-supervised deep learning approach to automate the classification of ten categories of rice leaf conditions, including seven diseases, two pests, and a healthy class. Field images were acquired using a custom solar-powered ESP32-CAM- based Internet of Things (IoT) edge node. To leverage unlabeled data and minimize manual annotation costs, the dataset was structured with 744 expert-labeled images and 10,249 unlabeled images. An initial Convolutional Neural Network (CNN) model was trained using the 744 expert-labeled images to generate pseudo-labels for the 10,249 unlabeled images. The combined labeled and pseudo-labeled dataset was subsequently partitioned into 80% training, 15% testing, and 5% validation sets to train the final transfer learning-based CNN and Support Vector Machine (SVM) models. Under the evaluated configurations, the CNN achieved higher performance than the SVM baseline, with an accuracy of 82.9%, precision of 83.0%, recall of 82.9%, F1-score of 0.829, and Matthews Correlation Coefficient (MCC) of 0.798, whereas the evaluated SVM configuration achieved an accuracy of 23.7%. These findings demonstrate the efficacy of combining low-cost IoT edge acquisition with semi-supervised deep learning for scalable, real-time crop disease monitoring in precision agriculture.**

***Keywords*—Semi-Supervised Learning (SSL); CNN; rice leaf disease; pest classification; IoT-based agriculture; deep learning; dataset**

I. Introduction

Rice is one of the main staple foods that significantly contributes to global food security, especially in Asian countries, including Indonesia. About 480 million tons of rice are produced each year, with China and India accounting for more than 50% of the total production [1], [2]. Rice is not only important for meeting the caloric needs of the people but also plays a role in supporting the economy and social stability, providing jobs for millions of farmers[3], [4]. Sometimes, rice plants get sick or are eaten by bugs, and this can make it hard to grow enough rice. Among these diseases and pests are Brown Spot (BS), Hispa (Pest), Bacterial Leaf Blight (BLB), Blast, Sheath Blight (SB), Dead Heart (Pest), Bacterial Panicle Blight (BPB), Tungro, and Bacterial Leaf Streak (BLS). Diseases and pests can lead to major losses and put the long-term success of rice farming at risk [5]–[9]. It is important to spot pest and disease problems early so you can act quickly and control them effectively, thereby minimizing losses and maintaining rice productivity amidst existing challenges [8], [10], [11]. Conventional methods for detecting diseases in rice generally require the expertise of trained specialists, are subjective, and cannot be widely or scalably applied. In the context of modern agriculture, this leads to delays in responding to disease outbreaks and results in significant losses for farmers [12], [13]. As a solution, the development of computer vision technology offers new opportunities to improve accuracy and efficiency in real-time disease detection, becoming an integral part of precision agriculture systems. The two main streams in this technology are: 1) the classic machine learning approach that uses handcrafted feature extraction, such as color, texture, and shape, accompanied by the use of classifiers like Support Vector Machine (SVM); and 2) the deep learning approach that allows models to learn features end-to-end without requiring manual intervention for feature extraction [14], [15]. Recent research shows that deep learning methods, especially convolutional neural networks (CNNs), can improve disease detection capabilities with high accuracy, reaching over 97% [16], [17]. This approach not only improves efficiency but also has the potential to save crops and enhance the sustainability of rice production by providing faster and more accurate detection of various diseases such as Leaf Blight, Brown Spot, and Blast [13], [18]. With the implementation of these techniques, it is expected that farmers will be better prepared to face disease threats, thereby maintaining food security more effectively[19], [20].

Against this practical context, the present study investigates a semi-supervised workflow that combines a limited expert-labeled set with a substantially larger pool of unlabeled field images for rice leaf disease and pest classification. The study integrates IoT-based field image acquisition using an ESP32-CAM node with CNN-based pseudo-label generation and compares the resulting CNN classifier with an SVM reference classifier across ten rice leaf condition classes. The objective is to evaluate a practical classification workflow for variable and imbalanced field imagery while reducing dependence on exhaustive expert annotation.

The remainder of this study is organized as follows. Section II reviews related work; Section III describes the materials and methods; Section IV presents the experimental results; Section V discusses the findings and deployment implications; and Section VI concludes the study and outlines future work.

*Corresponding author.

II. Background and Related Work

Semi-supervised learning (SSL) has recently emerged as an effective strategy for reducing the heavy annotation burden in agricultural computer vision, leveraging abundant unlabeled imagery alongside a small expert-labeled set. Reported applications span crop classification from satellite image time series [21], UAV-based maize seedling monitoring and counting [22], [23], multi-class weed detection and robotic weed management [24], [25], pixel-level and panoptic crop segmentation [26], [27], region-of-interest monitoring in vertical farms [28], and plant counting in aerial images under domain shift [29], with several pipelines approaching fully supervised accuracy while using only 5–30% of labeled data [21], [22], [23], [24], [25], [29], [30]. Within plant pathology specifically, SSL and self-training have improved disease and pest recognition: a lightweight self-training CNN achieved near-supervised accuracy for real-time jute leaf disease classification [31]; a semi-supervised one-stage detector raised mean average precision for maize leaf disease at low labeling ratios [32]; memory-based mutual learning produced robust pest detection from small labeled fractions [33]; and pseudo-labeling with consistency regularization (the Wheat Teacher detector) together with collaborative SSL enabled accurate wheat-head detection and wheat-lodging segmentation [34], [35].

A consistent finding across this literature is that the reliability of pseudo-labels governs SSL success; high-confidence pseudo-labels yield large gains, whereas unfiltered or erroneous labels propagate noise and can erode accuracy [32], [33], [36], [37]. Accordingly, several safeguards have been proposed: ambiguity-aware rejection of uncertain predictions [36], dynamic pseudo-label allocation with adaptive loss thresholds [34], iterative correction of noisy annotations [38], knowledge-distillation-based and one-shot pseudo-labeling demonstrated even for livestock video and animal identification [37], and depth-informed, almost-unsupervised pseudo-labeling for dense crop segmentation [39], [40]. Where the deployment domain differs from the training domain (different crops, growth stages, or imaging conditions), semi-supervised and semi-self-supervised domain adaptation, few-shot learning, and self-training improve cross-domain robustness [29], [30], [41], [42]. Collectively, these studies establish SSL with curated pseudo-labels rather than naive self-training as the practical route to scalable field analytics under limited supervision.

Despite this progress, two gaps motivate the present study. First, the very high accuracies, frequently exceeding 97%, reported for CNN-based rice-disease classification are predominantly obtained on curated, class-balanced secondary datasets captured under controlled conditions [43], [44]; images acquired in situ by low-cost IoT edge devices instead exhibit variable illumination, viewpoint, occlusion, and pronounced class imbalance, which substantially depress real-world performance consistent with the 82.9% accuracy obtained here on field-captured data. Second, most agricultural SSL studies target UAV or satellite imagery and crops other than rice, and few couple edge-based IoT image acquisition with a pseudo-label-driven semi-supervised workflow on a single, real field dataset. This study addresses both gaps by pairing a solar-powered ESP32-CAM acquisition node with a pseudo-labeling CNN that converts 10,249 unlabeled field images into usable training signal, and by benchmarking it against a handcrafted-feature SVM under realistic, imbalanced field conditions.

Research on plant disease classification using Support Vector Machine (SVM) has shown good accuracy, especially on limited and controlled datasets [12], [20]. SVM is often effective in classifying and detecting diseases because of its ability to separate data classes with an optimal decision boundary. However, a significant drawback of this approach is its dependence on the limited quantity and quality of training data, which does not always represent real-world conditions. On the other hand, the use of Convolutional Neural Networks (CNNs) in rice disease detection has shown superior performance, especially when using large datasets consisting of balanced labeled images from online repositories [45], [46]. Nevertheless, this approach tends to be less effective when applied to field datasets with more variable conditions. Previous studies generally focused only on balanced labeled data and did not consider challenges such as class imbalance and environmental variability caused by data acquisition through the Internet of Things (IoT). Current research clearly still lacks systematic comparisons between the two approaches, SVM and CNN, on real field datasets with unique characteristics, namely, limited labeled data (744) and abundant unlabeled data (10,249). This creates additional challenges for effective and efficient disease detection and highlights the need for further research to develop methodologies that can address these weaknesses in rice farming contexts [45], [46], [47]. This study makes a significant contribution to understanding and managing rice diseases by providing a comprehensive field dataset consisting of 744 labeled and 10,249 unlabeled data points, covering 10 classes of rice leaf conditions. The dataset was acquired through IoT technology, enabling more representative data collection aligned with dynamic field conditions [12], [46]. Furthermore, this research systematically compares traditional approaches (SVM with handcrafted feature extraction) and modern approaches (semi-supervised CNN) in limited data scenarios. This approach aims to enhance understanding of each method's strengths and weaknesses in a realistic context [13], [48]. Additionally, the study investigates the effectiveness of using unlabeled data by applying semi-supervised learning techniques, which aim to improve CNN model performance in rice disease detection. This method offers great potential to maximize available data and address the common issue of limited labeled data faced in previous research [49], [50]. Finally, the study provides recommendations for optimal architecture and strategies for field implementation based on IoT, intended to assist farmers in detecting and managing rice diseases efficiently and effectively. With support for integrating this technology, it is hoped that this research can contribute not only to increased agricultural yields but also to overall food security [14], [51], [52]

III. Materials and Methods

In this study, rice disease and pest classification on leaves was conducted using deep learning approaches based on Convolutional Neural Networks (CNN). The dataset used consists of 10,249 unlabeled images and 744 images labeled by experts. These images were obtained directly from the field using IoT devices that enable real-time monitoring of rice leaf conditions in agricultural environments. The types of diseases and pests classified include Brown Spot (disease), Hispa (pest),

Bacterial Leaf Blight (disease), Blast (disease), Sheath Blight (disease), Dead Heart (pest), Bacterial Panicle Blight (disease), Bacterial Leaf Streak (disease), and Tungro (disease), with varying numbers of images. Representative examples of the ten classes are shown in Fig. 1.

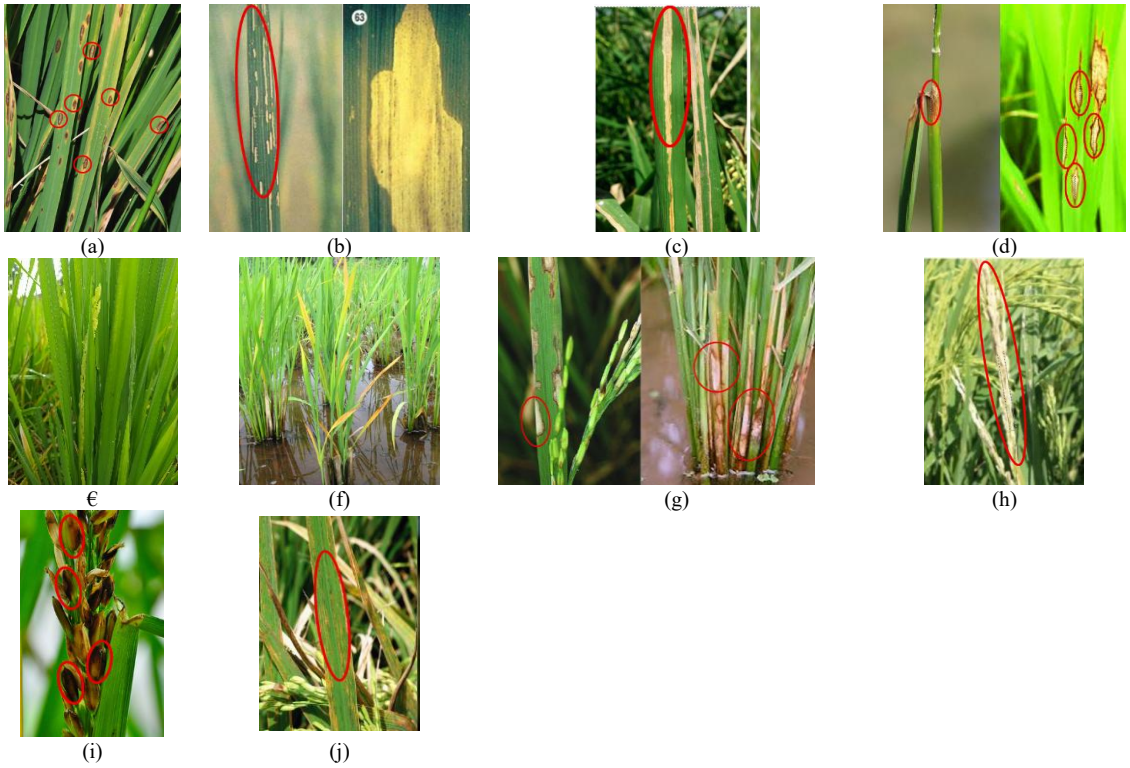


Fig. 1. Samples of the rice leaf diseases and pests dataset: (a) Brown Spot; (b) Hispa; (c) Bacterial Leaf Blight; (d) Blast; (e) Normal; (f) Tungro; (g) Sheath Blight; (h) Dead Heart; (i) Bacterial Panicle Blight; and (j) Bacterial Leaf Streak.

The collected data comprise images of rice leaves obtained from the field using IoT devices, totaling 10,249 unlabeled images and 744 labeled by experts. From this data, the dataset is divided into two groups: training data (learning dataset) and performance data (performance dataset). In the training dataset, the process is carried out gradually until a trained model is produced. The model is then tested to measure accuracy using performance data. If the accuracy meets the standard, the model is finalized. If not, other methods are applied until satisfactory performance is achieved. This methodology aims to continuously optimize the rice leaf disease and pest classification model by improving accuracy. The overall research workflow is summarized in Fig. 2.

This flowchart diagram illustrates the research methodology for classifying rice leaf diseases and pests using CNN and SVM. The process begins with data collection, preprocessing (normalization and augmentation), followed by data splitting. Two models are selected, CNN and SVM, and trained using the data. A semi-supervised approach is used for unlabeled data. After training, the models are evaluated based on accuracy, precision, recall, and F1 score, and then the best model is used to predict unlabeled rice leaf images. The final stage involves applying the model for real-time field prediction of diseases and pests.

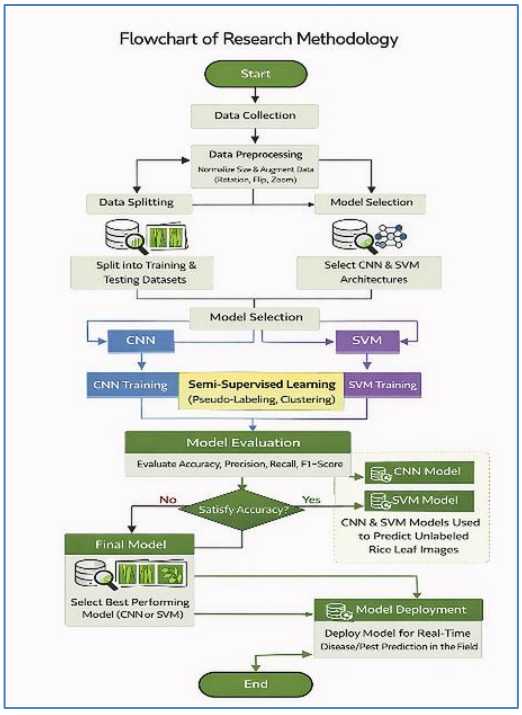


Fig. 2. Flowchart of research methodology.

A. Data Acquisition and IoT Setup

The ESP32-CAM module attracts attention in agricultural IoT because it is inexpensive, has integrated Wi-Fi, and has enough resolution to monitor plant diseases [53]. Combined with ultrasonic sensors for distance measurement, servo motors for precise positioning, and GPS modules for geotagging, the ESP32-CAM-based system offers an affordable solution for automatic field image acquisition [54]. Table I summarizes the specifications of the IoT components used.

TABLE I. IOT NODE COMPONENT SPECIFICATIONS

Components	Specifics	Function
ESP32-CAM	ESP32-S dual-core 240MHz, 4MB PSRAM, Wi-Fi	Main controller and camera
Camera OV2640	2MP, resolution 1600×1200	Filming
Sensor HC-SR04	Distance 2- 400 cm, accuracy ±3mm	Distance measurement
Servo MG90S (2)	Torque 2.2 kg·cm, 0-180°	Pan and tilt control
GPS NEO-6M	Accuracy 2.5m, update rate 5Hz	Geotagging
Buzzer 5V	85dB, 10cm	Field operator alert
Solar Panel	20W polycrystalline	Main resources
Battery	12V 12Ah lead-acid	Power storage
Regulator LM2596	Step-down 12V to 5V	Power regulation
Module TP4056	Charging with protection	Battery management

Fig. 3 illustrates the hardware assembly and pin connections of the IoT node. The ESP32-CAM is connected to the GPIO pins, and Fig. 4 shows the process of capturing images of rice leaves in the field.



Fig. 3. IoT device.



Fig. 4. Photographing in the rice field.

B. Dataset and Ground Truth:

In this study, the dataset [55] used is divided into two main parts, namely Labeled Data and Unlabeled Pool. The Labeled Data consists of 744 images that were verified by three experts from the Faculty of Agriculture, Universitas Jenderal Soedirman (UNSOED), Purwokerto, Indonesia: Prof. Ir. Totok Agung Dwi Haryanto, M.P., Ph.D. (Plant Breeding); Woro Sri Suharti, S.P., M.P., Ph.D. (Plant Pathology); and Dr. Endang Mugiastuti, S.P., M.P. (Plant Pathology). Each image is classified into a category of disease or pest on rice leaves, with details available in Table II.

TABLE II. DISTRIBUTION OF RICE LEAF DISEASE CLASSES IN THE LABELED DATASET

Disease Class	Number of Images	Class classification
Bacterial Leaf Blight	85	Diseases
Brown Spot	120	Diseases
Blast	83	Diseases
Sheath Blight	59	Diseases
Tungro	50	Diseases
Normal	72	Normal
Dead Heart	56	Pests and Diseases
Bacterial Panicle Blight	53	Diseases
Bacterial Leaf Streak	50	Diseases
Hispa	116	Pests and Diseases
Quantity	744	

Meanwhile, the Unlabeled Pool contains 10,249 raw images taken directly from the field. These images do not yet have expert-assigned class labels and cover a variety of conditions, including different rice plant varieties, growth stages, lighting conditions, and viewing angles. In the semi-supervised workflow, this pool is processed by the initial teacher CNN to obtain predicted class labels. The prediction output is then converted into the class-label field and combined with the expert-labeled data before the final train/test/validation partition is created. Representative samples from the unlabeled image pool are shown in Fig. 5.



Fig. 5. Unlabeled data.

C. Pre-Processing Data

In the data preprocessing stage, all images in the dataset are resized to suit the needs of each model. Images for the Convolutional Neural Network (CNN) model are resized to 224x224 pixels, while images for feature extraction with Support Vector Machine (SVM) are reduced to 227x227 pixels.

D. Convolutional Neural Network

Semi-supervised CNN is a machine learning approach that combines labeled and unlabeled data to train object detection and classification models in images. The model is trained with a small amount of labeled data to learn initial representations, then optimized with unlabeled data through the creation of pseudo-labels. The main challenge is errors in pseudo-labels, which can reduce accuracy if not handled properly [56].

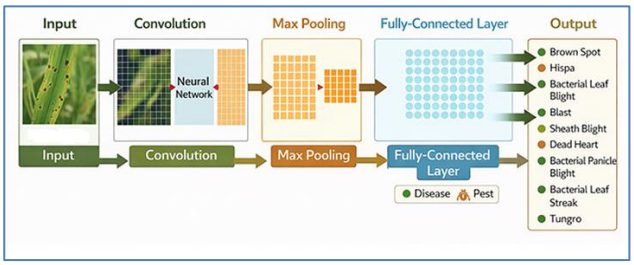


Fig. 6. CNN workflow.

Fig. 6 illustrates the process of classifying rice leaf diseases and pests using a CNN, starting from an input image measuring 224x224 pixels. The image is processed through convolution to extract features such as texture and shape, then through max pooling to reduce dimensions while preserving key features. The features are then classified as Brown Spot, Hispa, and others via a fully connected layer, enabling the system to automatically detect rice plant diseases and pests, thereby assisting agricultural management. CNN utilizes the convolution process to divide the image into small parts, extract numerical features through encoding, and use down-sampling (max pooling) to reduce the array size, which speeds up training and decreases layer complexity. At the end of the convolutional network, there is an evaluation section used to measure the system's performance. The model is optimized using the categorical cross-entropy loss, while its classification performance is assessed using Accuracy, Precision, Recall, F1-score, and the Matthews Correlation Coefficient (MCC), as defined in Equations (1)–(6). Here, TP, TN, FP, and FN denote the true positives, true negatives, false positives, and false negatives, respectively; N is the number of observations and K is the number of classes.

$$\text{Accuracy} = \frac{\sum_{k=1}^{K} \text{TP}_k}{N} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F}_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{5}$$

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log(\hat{y}_{i,k}) \tag{6}$$

The initial preprocessing stage of rice leaf data involves configuring and collecting data to train a CNN model. Out of 10,993 IoT images, 744 are validated and labeled by plant experts across 10 categories, while 10,249 images form the unlabeled pool used for semi-supervised learning. After training on the 744 expert-labeled images, the initial CNN is applied to the 10,249 unlabeled images in the Predictions Data widget. The predicted class output is passed through the Edit Domain ("Edit Label") widget to standardize it as the target class variable, producing the pseudo-labeled pool used in the subsequent combined-data stage. For comparison, the study also employs Support Vector Machine (SVM) as a conventional reference classifier using handcrafted features and the fixed configuration listed in Table V, including a sigmoid kernel. Evaluation is performed using confusion matrices and metrics including accuracy, precision, recall, F1-score, and MCC.

To preserve separation in the final evaluation stage, the expert-labeled and pseudo-labeled data are concatenated before the final-model split. The first stratified Data Sampler, using a fixed seed, allocates 80% of the combined pool to training. From the remaining 20%, a second stratified sampler allocates 75% to testing, corresponding to 15% of the complete pool, while the remaining 5% is reserved for validation. Consequently, the final CNN and SVM are trained only on the 80% training partition, while the 15% test and 5% validation partitions remain non-overlapping hold-out subsets.

E. Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm used in classification and regression analysis. This algorithm belongs to the category of supervised learning, which means it requires labeled data for training. SVM is well-known for its ability to find the optimal hyperplane that separates classes in the feature space with the maximum margin between data points from two different classes. The basis of SVM is the search for a hyperplane that separates two classes of data. This hyperplane can be viewed as a line (in two dimensions), or a plane (in three dimensions), and the model aims to maximize the margin, which is the distance between the closest data points of each class. The data points closest to the hyperplane are called support vectors, which are crucial points because they determine the position of the hyperplane [56], [57].

IV. RESULT

The data preparation process is implemented to ensure consistency across the entire dataset. A total of 10,993 rice leaf images are used, consisting of 744 labeled images validated by experts and 10,249 unlabeled field images taken through IoT devices (Fig. 8). The labeled dataset includes 10 different classes, including 7 disease categories (bacterial leaf blight, bacterial leaf streak, bacterial panicle blight, blast, brown spot, sheath blight, and tungro), 2 pest categories (dead heart and hispa), and 1 healthy class (normal). The 744 expert-labeled images are first used to train the initial CNN model, which subsequently generates predictions for the 10,249 unlabeled field images. After label assignment, the resulting data are combined with the expert-labeled dataset for final model development. The combined dataset is then partitioned using stratified sampling into 80% training, 15% testing, and 5% validation subsets. The reported performance metrics therefore correspond to this predefined stratified hold-out evaluation.

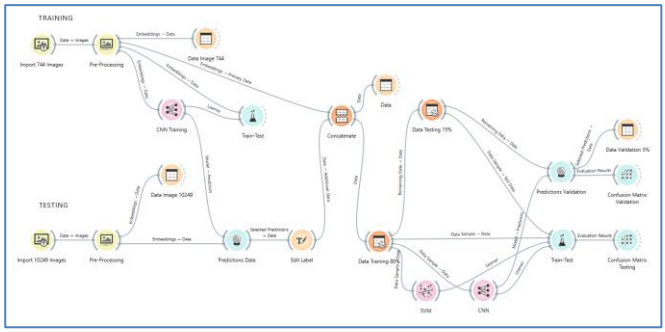


Fig. 7. Result of model development using graphical data mining application.

Based on the workflow in Fig. 6, the classification procedure consists of an initial teacher-model and pseudo-labeling stage followed by final model development and evaluation. In the first stage, the 744 expert-validated images are processed through pre-processing and feature extraction to train the initial CNN, which is then used to predict labels for the 10,249 unlabeled field images. The prediction output is standardized as the class variable through the Edit Label step and then concatenated with the expert-labeled data. Only after this label-assignment stage is the combined dataset divided into 80% training data, 15% testing data, and 5% validation data. The final CNN and SVM models are evaluated using confusion matrices and performance measures including accuracy, precision, recall, F1-score, and MCC (Fig. 7).

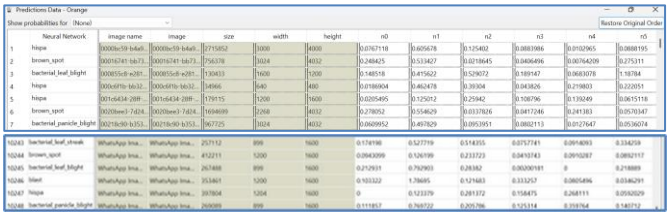


Fig. 8. Image labeling process of 10,993 images.

A. Convolutional Neural Network

CNN Training serves as an initial model that functions as a 'teacher' before the final model is created. Starting with the expert-labeled rice leaf images, the CNN learns visual patterns from the disease and pest classes and is then used in the Predictions Data widget to assign pseudo-labels to the unlabeled image pool. The predicted class output is standardized through the Edit Label widget and combined with the expert-labeled data through the Concatenate widget. The combined pool is then divided into 80% training, 15% testing, and 5% validation using the sequential stratified samplers. The 5% validation partition contains 549 images and is used for the confusion-matrix analysis in Fig. 9, whereas the 15% test partition is used for the summary performance metrics reported in Table IV. CNN Training therefore functions as the teacher stage for expanding the available training signal before the final CNN and SVM evaluation.

Details of the training configuration for the CNN model from the Graphical Data Mining Application are shown in Table III.

TABLE III. DETAILS OF THE TRAINING LAYERS FOR THE CNN

No	Parameters	Value
1	Neuron in Hidden Layer	200,100
2	Activation	ReLu
3	Solver/Optimizer	Adam
4	Regulation optimizer	0,001
5	Iteration	500

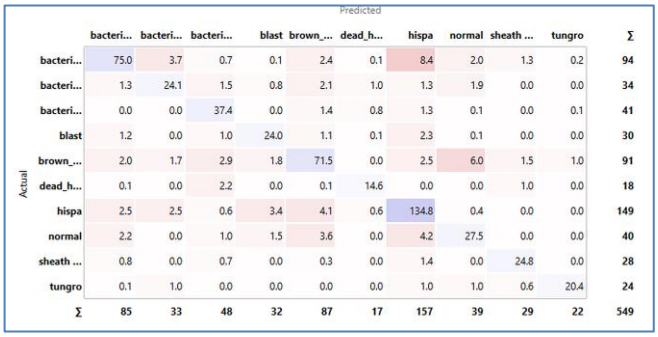


Fig. 9. CNN validation confusion matrix 5%.

The confusion matrix shown in this analysis illustrates the detailed performance of the Convolutional Neural Network (CNN) model in classifying ten categories of rice leaf conditions, including seven types of diseases, two types of pests, and one healthy class. In the matrix, the actual labels are shown by rows and the model predictions by columns, while the main diagonal represents correctly assigned class membership. The matrix is calculated on the 5% validation partition of the combined dataset, comprising 549 images. This validation partition is distinct from the test partition used for the summary performance metrics reported in Table IV.

The CNN confusion matrix indicates comparatively strong class-level recall for Hispa (90.5%) and Bacterial Leaf Streak (91.2%), followed by Sheath Blight (88.6%) and Dead Heart (81.1%). In contrast, Tungro showed the lowest recall at 52.2%, with a notable tendency to be confused with Sheath Blight. Bacterial Panicle Blight and Brown Spot achieved class-level recall of 65.1% and 73.7%, respectively. The matrix also indicates overprediction of Hispa and underprediction of

Tungro. Overall, most classes showed comparatively strong recall, while some classes, particularly Tungro, remained more challenging. These class-wise confusion patterns indicate that visually similar symptoms continue to be an important source of classification error and motivate further improvements through additional representative samples and data augmentation.

TABLE IV. RESULTS OF THE DEVELOPED MODEL CNN

No	Result	Value on Data Testing (%)
1	Accuracy	82,9
2	Precision	83
3	Recall	82,9
4	F1-Score	0,829
5	MCC	0.798

Table IV summarizes the performance of the Convolutional Neural Network (CNN) model for classifying 10 categories of rice leaf conditions (7 diseases, 2 pests, and 1 healthy class) on the final combined-data test partition defined by the 80% training, 15% testing, and 5% validation split. The separate 5% validation partition, comprising 549 images, is used for the confusion-matrix analysis in Fig. 9. The reported metrics include accuracy, precision, recall, F1-score, MCC, and AUC. The CNN achieved an AUC of 0.985, indicating strong discriminative capability across the evaluated classes.

B. Support Vector Machine

Pre-processing methods for the SVM and CNN models follow the same scheme as the workflow diagram that shows the data splitting details. The data is grouped into ten folders according to the rice disease class names, including Bacterial Leaf Blight and Blast. Images are converted into a format readable by the model after being resized using the pre-processing widget in Orange, and visual features are extracted through an Image Embedding method based on Inception v3. For initial CNN training, 744 labeled images are used, generating predictions for 10,249 unlabeled images through pseudo-labeling. All labeled data are then combined into approximately 11,000 images. For final model development, the combined dataset is partitioned into training (80%), testing (15%), and validation (5%) subsets used by both the CNN and SVM evaluation workflows in Orange Data Mining. The SVM shows lower performance than the CNN under the evaluated configuration, as reflected by CA, F1, Precision, Recall, and MCC, with higher prediction errors especially in visually similar classes such as Blast and Brown Spot. The performance of both models can be viewed through the Confusion Matrix. Details of the SVM configuration from the Graphical Data Mining Application in Fig. 10 are shown in Table V.

TABLE V. SVM MODEL CONFIGURATION

No	Parameters	Value
1	SVM Cost	1,0
2	Regression loss epsilon	0,10
3	Kernel	Sigmoid
4	SVM optimizer tolerance	0,0010
5	SVM Iteration Limit	100



Fig. 10. SVM validation confusion matrix 5%.

The SVM confusion matrix shows limited class-wise performance across the 10 rice leaf categories. Hispa achieved the highest class-level recall at 42.8%, while several other classes showed substantially lower recall, and Bacterial Leaf Blight and Blast were rarely classified correctly. These patterns are consistent with the overall SVM results in Table VI and indicate that the evaluated SVM configuration had difficulty separating visually similar disease and pest classes.

TABLE VI. RESULTS OF THE DEVELOPMENT MODEL SVM

No	Result	Value on Data Testing (%)
1	Accuracy	23,7
2	Precision	27,6
3	Recall	23,7
4	F1-Score	25,5
5	MCC	15,1

This study compares CNN and SVM for rice leaf image classification. The SVM is trained and evaluated using the same final combined-data split described above, with 80% allocated to training, 15% to testing, and 5% to validation. On the test partition, the SVM achieved an AUC of 0.761 and an accuracy of 23.7%. The F1-Score of 0.255 indicates prediction imbalance, and the MCC of 0.151 reflects limited agreement between predicted and actual classes under the evaluated configuration.

V. DISCUSSION

A. Performance Comparison of CNN and SVM Classifiers

This study compares the CNN and SVM implementations evaluated for classifying rice leaf diseases and pests. The CNN achieved an AUC of 0.985, classification accuracy (CA) of 82.9%, F1-score of 0.829, precision of 83.0%, recall of 82.9%, and an MCC of 0.798. Under the specified SVM configuration, the corresponding testing accuracy was 23.7% with an MCC of 0.151. The SVM confusion matrix shows frequent misclassification, particularly for disease classes with overlapping visual symptoms such as Blast and Brown Spot. Accordingly, the comparison in this study is interpreted as the performance of the two implemented configurations on the same field-acquired dataset, rather than as an exhaustive assessment of all possible SVM parameterizations.

B. Annotation and Computational Trade-Offs

While the CNN model offers high accuracy, it requires a substantial labeled dataset to initialize training. In this study, the

training of the initial "teacher" CNN utilized 744 expert-labeled images to generate pseudo-labels for the 10,249 unlabeled images. In terms of workflow constraints, expert manual annotation of the entire 10,993-image dataset represents a significant bottleneck in terms of time and cost. The SVM model presents a simpler alternative that does not require GPU acceleration for inference. However, traditional SVMs face scaling limitations as the size of the training feature space grows, leading to high training complexity, even though their final testing (inference) time remains negligible. Under the configurations evaluated in this study, the semi-supervised CNN provided the stronger classification performance. Future work may broaden the benchmark by examining alternative SVM parameterizations together with other classifiers such as Random Forest, K-Nearest Neighbors (KNN), and additional deep learning architectures.

C. Impact of Semi-Supervised Learning on Class Imbalance

A key finding in this study is the interaction between Semi-Supervised Learning (SSL) and the class imbalance inherent in the labeled dataset (Table II). The majority class (Brown Spot) contains 120 images, whereas minority classes (Tungro and Bacterial Leaf Streak) contain only 50 images each. In self-training workflows, this imbalance presents a risk of confirmation bias because the pseudo-label assignments are generated by a model initialized from the same imbalanced expert-labeled distribution. As a result, class preferences learned during the teacher stage can be propagated into the pseudo-labeled pool and subsequently influence the combined-data training stage.

This behavior provides a plausible explanation for the lower class-level recall of Tungro (52.2%) and its recurring confusion with Sheath Blight, which shares visually similar yellowing symptoms. Conversely, Hispa tended to be overpredicted. These patterns are consistent with the general risk that pseudo-labeling can reinforce initial class preferences when visually overlapping classes are present. Future implementations could therefore consider class-aware pseudo-label selection and human-in-the-loop review of uncertain cases before final model retraining.

D. Proposed Field Deployment Architecture and Real-Time Farmer Notification Pipeline

To illustrate how the classification framework could be extended into a practical precision agriculture service, a real-time data integration and notification architecture is proposed. In this architecture, the ESP32-CAM IoT edge node deployed in the field would perform image acquisition and geotagging via the NEO-6M GPS module, and the captured images could be transmitted to a cloud-based server through a local Wi-Fi gateway or cellular network such as 4G/5G. The computational task of running the trained transfer learning CNN model could then be handled by the cloud server. When a disease or pest is detected, the classification result, GPS coordinates, and timestamp could be stored in a Firebase Realtime Database.

In the proposed farmer-facing layer, database updates could trigger an automated webhook to send a push notification or instant message through services such as the WhatsApp Business API. The notification could link to a lightweight mobile dashboard displaying:

- The classified disease or pest type.
- A confidence score and the captured leaf image.
- The exact geographic location of the affected crop to facilitate targeted pesticide application or physical intervention.

Such a cloud-to-mobile pipeline would allow farmers to access the classification results without requiring specialized technical knowledge or high-performance local computing devices, and could support rapid, localized responses to pest and disease outbreaks.

VI. Conclusion

This study investigated rice leaf disease and pest classification using a semi-supervised Convolutional Neural Network (CNN) workflow and a Support Vector Machine (SVM) reference classifier. The CNN used 744 expert-labeled images to generate labels for 10,249 unlabeled field images, after which the combined dataset was used for final model development. Under the configurations evaluated in this study, the CNN achieved 82.9% accuracy compared with 23.7% for the SVM baseline, while also providing a practical mechanism for exploiting the large unlabeled image pool. These results support the use of the semi-supervised CNN workflow for the field-image classification setting examined here. Future work may extend the comparison to additional classifier configurations and architectures, apply data-augmentation strategies, and explore lighter CNN models and more advanced semi-supervised techniques for resource-limited deployment. Future studies may complement this hold-out evaluation with repeated or cross-validation-based assessment across additional field datasets.

Acknowledgment

The author would like to express sincere gratitude to Universitas Ahmad Dahlan for providing academic support and guidance during the preparation of this research. The author also gratefully acknowledges Universitas Amikom Purwokerto for providing scholarship support during the doctoral study. Appreciation is extended to the doctoral program supervisors and lecturers for their valuable insights, constructive feedback, and encouragement throughout the development of this study. This work was conducted as part of the author's doctoral academic activities in the field of deep learning and IoT-based agricultural disease detection.

References

[1] Y. M. Alsakar, N. A. Sakr, and M. Elmogy, "An Enhanced Classification System of Various Rice Plant Diseases Based on Multi-Level Handcrafted Feature Extraction Technique," Scientific Reports, 2024, https://doi.org/10.1038/s41598-024-81143-1.

[2] S. Akter, R. I. Sumon, H. Ali, and H. C. Kim, "Utilizing Convolutional Neural Networks for the Effective Classification of Rice Leaf Diseases Through a Deep Learning Approach," Electronics, 2024, https://doi.org/10.3390/electronics13204095.

[3] V. R. T. Sai, "Rice Leaf Disease Detection Using CNN," International Journal for Research in Applied Science and Engineering Technology, 2023, https://doi.org/10.22214/ijraset.2023.54638.

[4] M. M. Hasan, T. Rahman, A. Uddin, S. M. Galib, and ..., "Enhancing rice crop management: Disease classification using convolutional neural networks and mobile application integration," Agriculture, 2023, [Online]. Available: https://www.mdpi.com/2077-0472/13/8/1549.

[5] R. Sharma, A. Singh, N. Z. Jhanjhi, and ..., "Plant Disease Diagnosis and Image Classification Using Deep Learning," … Materials &Continua, 2022, [Online]. Available: https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC-71-2/TSP_CMC_20017/TSP_CMC_20017.pdf.

[6] Abdul Azis, Abdul Fadlil, and Tole Sutikno, "Optimization of Convolutional Neural Network (CNN) Using Transfer Learning for Disease Identification in Rice Leaf Images," Jurnal E-Komtek (Elektro-Komputer-Teknik), vol. 8, no. 2, pp. 504–515, Dec. 2024, https://doi.org/10.37339/e-komtek.v8i2.2132.

[7] Abdul Azis, Abdul Fadlil, and Tole Sutikno, "Real-Time Rice Leaf Disease Diagnosis: A Mobile CNN Application with Firebase Integration," Jurnal Teknik Informatika (Jutif), vol. 6, no. 3, pp. 1469–1484, Jun. 2025, https://doi.org/10.52436/1.jutif.2025.6.3.4452.

[8] V. Agarwal, M. Kshirsagar, R. Jain, and S. Jain, "Smart Mobile System for Detection and Classification of the (Oryza Sativa) Rice Plant Disease Using Deep Learning and Transfer Learning," International Journal for Research in Applied Science and Engineering Technology, 2022, https://doi.org/10.22214/ijraset.2022.46554.

[9] T. W. Utami, "Implementation and Comparative Analysis of CNN and Transfer Learning Models (EfficientNetB0, MobileNetV2, and ResNet50) for Rice Leaf Disease Detection Based on Digital Images," Journal of Applied Informatics and Computing, 2025, https://doi.org/10.30871/jaic.v9i6.11616.

[10] F. N. Fiqri, S. Setyaningsih, and A. Saepulrohman, "Rice Disease Image Classification using MobileNetV2 Pretrained Model with Attention Visualization using Gradient-weighted Class Activation Mapping (Grad-CAM)," in 2023 6th International Conference on Information and Communications Technology (ICOIACT), 2023, pp. 367–371, https://doi.org/10.1109/ICOIACT59844.2023.10455786.

[11] P. A. S. Rani and N. S. Singh, "Paddy leaf symptom-based disease classification using deep CNN with ResNet-50," … Journal of Advanced Science Computing and …, 2022, [Online]. Available: https://www.ijasce.org/index.php/IJASCE/article/view/83.

[12] K. K. K and KE, "Detection of rice plant disease using AdaBoostSVM classifier," Agronomy journal, 2022, https://doi.org/10.1002/agj2.21070.

[13] N. Thanuboddi and U. R. Nelakuditi, "Hybrid deep learning for smart paddy disease diagnosis using self supervised hierarchical reconstruction and attention based temporal analysis," Scientific Reports, vol. 15, no. 1, p. 34917, 2025, https://doi.org/10.1038/s41598-025-18672-w.

[14] C. Subbarayudu and M. Kubendiran, "An Automated Hybrid Deep Learning Framework for Paddy Leaf Disease Identification and Classification," Scientific Reports, 2025, https://doi.org/10.1038/s41598-025-08071-6.

[15] P. Prakash and M. Srivenkatesh, "Impact and Analysis of Disease Spread in Paddy Crops Using Environmental Factors With the Support of X-Step Algorithm," International Journal of Advanced Computer Science and Applications, 2023, https://doi.org/10.14569/ijacsa.2023.0140570.

[16] V. R. Nagarajan, "Precision Farming with AI: An Integrated Deep Learning Solution for Paddy Leaf Disease Monitoring." [Online]. Available: www.ijacsa.thesai.org.

[17] S. Tepdang and K. Chamnongthai, "Boundary-Based Rice-Leaf-Disease Classification and Severity Level Estimation for Automatic Insecticide Injection," Applied Engineering in Agriculture, vol. 39, no. 3, pp. 367–379, 2023, https://doi.org/https://doi.org/10.13031/aea.15257.

[18] R. Wang et al., "Occurrence and Integrated Control of Major Rice Diseases in China," New Plant Protection, 2025, https://doi.org/10.1002/npp2.70004.

[19] T. H. Nhut, V. P. T. Trung, C. M. Khoi, N. P. H. Khang, and N. C. Nhan, "An AIoT-Based System for Efficient Classification of Rice Leaf Diseases†," IEEJ Transactions on Electrical and Electronic Engineering, vol. 20, no. 11, pp. 1875–1881, Nov. 2025, https://doi.org/https://doi.org/10.1002/tee.70163.

[20] S. Aggarwal et al., "Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improvise Agro-Business," Scientific Programming, vol. 2022, no. 1, p. 1757888, Jan. 2022, https://doi.org/https://doi.org/10.1155/2022/1757888.

[21] X. Hu et al., "Hierarchical grouped contrastive representation learning for crop classification in satellite image time series," Journal of Applied Remote Sensing, vol. 19, no. 3, p. 038503, Jul. 2025, https://doi.org/10.1117/1.JRS.19.038503.

[22] R. Yang et al., "Integrating UAV remote sensing and semi-supervised learning for early-stage maize seedling monitoring and geolocation," Plant Phenomics, vol. 7, no. 1, p. 100011, 2025, https://doi.org/https://doi.org/10.1016/j.plaphe.2025.100011.

[23] X. Xu et al., "Maize Seedling Leave Counting Based on Semi-Supervised Learning and UAV RGB Images," Sustainability, vol. 15, no. 12, 2023, https://doi.org/10.3390/su15129583.

[24] J. Li, D. Chen, X. Yin, and Z. Li, "Performance evaluation of semi-supervised learning frameworks for multi-class weed detection," Frontiers in Plant Science, vol. Volume 15-2024, 2024, https://doi.org/10.3389/fpls.2024.1396568.

[25] W. Zhang, Z. Miao, N. Li, C. He, and T. Sun, "Review of Current Robotic Approaches for Precision Weed Management," Current Robotics Reports, vol. 3, no. 3, pp. 139–151, 2022, https://doi.org/10.1007/s43154-022-00086-5.

[26] J. Lee et al., "A pixel-level coarse-to-fine image segmentation labelling algorithm," Scientific Reports, vol. 12, no. 1, p. 8672, 2022, https://doi.org/10.1038/s41598-022-12532-7.

[27] P. Zimmer, M. Halstead, and C. Mccool, Panoptic One-Click Segmentation: Applied to Agricultural Data. 2023.

[28] Y. Hwang, S. Lee, T. Kim, K. Baik, and Y. Choi, "Crop Growth Monitoring System in Vertical Farms Based on Region-of-Interest Prediction," Agriculture, vol. 12, no. 5, 2022, https://doi.org/10.3390/agriculture12050656.

[29] J. Rodriguez-Vazquez, M. Fernandez-Cortizas, D. Perez-Saura, M. Molina, and P. Campoy, "Overcoming Domain Shift in Neural Networks for Accurate Plant Counting in Aerial Images," Remote Sensing, vol. 15, no. 6, 2023, https://doi.org/10.3390/rs15061700.

[30] Y. Li and X. Chao, "Semi-supervised few-shot learning approach for plant diseases recognition," Plant Methods, vol. 17, no. 1, p. 68, 2021, https://doi.org/10.1186/s13007-021-00770-1.

[31] M. Jannat et al., "Real-time jute leaf disease classification using an explainable lightweight CNN via a supervised and semi-supervised self-training approach," Frontiers in Plant Science, vol. Volume 16-2025, 2025, https://doi.org/10.3389/fpls.2025.1647177.

[32] J. Liu, Y. Hu, Q. Su, J. Guo, Z. Chen, and G. Liu, "Semi-Supervised One-Stage Object Detection for Maize Leaf Disease," Agriculture, vol. 14, no. 7, 2024, https://doi.org/10.3390/agriculture14071140.

[33] J. Zhou et al., "Mutual learning with memory for semi-supervised pest detection," Frontiers in Plant Science, vol. Volume 15-2024, 2024, https://doi.org/10.3389/fpls.2024.1369696.

[34] R. Zhang, M. Yao, Z. Qiu, L. Zhang, W. Li, and Y. Shen, "Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods," Agriculture, vol. 14, no. 2, 2024, https://doi.org/10.3390/agriculture14020327.

[35] H. Zhi, B. Yang, and Y. Zhu, "Collaborative Wheat Lodging Segmentation Semi-Supervised Learning Model Based on RSE-BiSeNet Using UAV Imagery," Agronomy, vol. 13, no. 11, 2023, https://doi.org/10.3390/agronomy13112772.

[36] T.-C. Pham, T.-N. Nguyen, and V.-D. Nguyen, "Ambiguity-aware semi-supervised learning for leaf disease classification," Scientific Reports, vol. 15, no. 1, p. 14070, 2025, https://doi.org/10.1038/s41598-025-95849-3.

[37] Y. Qiao et al., "One-Shot Learning with Pseudo-Labeling for Cattle Video Segmentation in Smart Livestock Farming," Animals, vol. 12, no. 5, 2022, https://doi.org/10.3390/ani12050558.

[38] J. Dong, A. Fuentes, S. Yoon, H. Kim, and D. S. Park, "An iterative noisy annotation correction model for robust plant disease detection," Frontiers in Plant Science, vol. Volume 14-2023, 2023, https://doi.org/10.3389/fpls.2023.1238722.

[39] R. E. P. Ferreira, Y. J. Lee, and J. R. R. Dórea, "Using pseudo-labeling to improve performance of deep neural networks for animal identification," Scientific Reports, vol. 13, no. 1, p. 13875, 2023, https://doi.org/10.1038/s41598-023-40977-x.

[40] S. Cao et al., "The blessing of Depth Anything: An almost unsupervised approach to crop segmentation with depth-informed pseudo labeling," Plant Phenomics, vol. 7, no. 1, p. 100005, 2025, https://doi.org/https://doi.org/10.1016/j.plaphe.2025.100005.

[41] G. Tüysüzoğlu, D. Birant, and V. Kıranoglu, "Soil Temperature Prediction via Self-Training: Izmir Case," Journal of Agricultural Sciences, vol. 28, no. 1, pp. 47–62, 2022, https://doi.org/10.15832/ankutbd.775847.

[42] A. Ghanbari, G. H. Shirdel, and F. Maleki, "Semi-Self-Supervised Domain Adaptation: Developing Deep Learning Models with Limited Annotated Data for Wheat Head Segmentation," Algorithms, vol. 17, no. 6, 2024, https://doi.org/10.3390/a17060267.

[43] P. Kumar and V. Nagarajan, "Precision Farming With AI: An Integrated Deep Learning Solution for Paddy Leaf Disease Monitoring," International Journal of Advanced Computer Science and Applications, 2024, https://doi.org/10.14569/ijacsa.2024.0150783.

[44] S. Tepdang and K. Chamnongthai, "Boundary-Based Rice-Leaf-Disease Classification and Severity Level Estimation for Automatic Insecticide Injection," Applied Engineering in Agriculture, 2023, https://doi.org/10.13031/aea.15257.

[45] S. Kokila, S. Abirami, and D. Nagaraju, "Image-Based Deep Learning Method for Detecting Diseases in Rice Plants," Engineering, Technology & Applied Science Research, vol. 15, no. 6, pp. 28943–28949, 2025, https://doi.org/10.48084/etasr.11930.

[46] S. B. Ahmed, S. F. Ali, and A. Z. Khan, "On the Frontiers of Rice Grain Analysis, Classification and Quality Grading: A Review," IEEE Access, vol. 9, pp. 160779–160796, 2021, https://doi.org/10.1109/ACCESS.2021.3130472.

[47] M. Chithambarathanu and M. K. Jeyakumar, "Survey on crop pest detection using deep learning and machine learning approaches," Multimedia Tools and Applications, vol. 82, no. 27, pp. 42277–42310, 2023, https://doi.org/10.1007/s11042-023-15221-3.

[48] M. Amudha and B. K, "Hybrid Paddy disease classification using optimized statistical feature based transformation technique with explainable AI," Environmental Research Communications, vol. 7, no. 7, p. 075017, 2025, https://doi.org/10.1088/2515-7620/adecec.

[49] S. Bondre and D. Patil, "Recent Advances in Agricultural Disease Image Recognition Technologies: A Review," Concurrency and Computation Practice and Experience, 2023, https://doi.org/10.1002/cpe.7644.

[50] M. Chithambarathanu and M. K. Jeyakumar, "Survey on Crop Pest Detection Using Deep Learning and Machine Learning Approaches," Multimedia Tools and Applications, 2023, https://doi.org/10.1007/s11042-023-15221-3.

[51] V. R. Nagarajan, "Precision Farming with AI: An Integrated Deep Learning Solution for Paddy Leaf Disease Monitoring." [Online]. Available: www.ijacsa.thesai.org.

[52] S. K. Upadhyay and A. K. Dwivedi, "Rice Plant Disease Diagnosis Using SqueezeNet and Deep Transfer Learning," Journal of Phytopathology, vol. 173, no. 3, p. e70092, May 2025, https://doi.org/https://doi.org/10.1111/jph.70092.

[53] R. K. Kodali, S. Yerroju, and L. Boppana, "ESP32-CAM Based Surveillance System Using IoT," in 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 1–5, https://doi.org/10.1109/TENSYMP50017.2020.9230950.

[54] A. Kumar, R. Singh, and S. Patel, "IoT-Based Smart Agriculture Monitoring System Using ESP32 and Sensor Integration," in 2021 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), 2021, pp. 1–6, https://doi.org/10.1109/ICSCEE50312.2021.9498214.

[55] A. AZIS, A. Fadlil, and T. Sutikno, "Dataset for: Semi-Supervised Deep Learning for Rice Leaf Disease and Pest Classification via IoT ." Zenodo, Jun. 2026, https://doi.org/10.5281/zenodo.20838112.

[56] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000.

[57] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995, https://doi.org/10.1007/BF00994018.

Constraint-Aware Automated Daily Meal Planning: A Controlled Comparison of Genetic Algorithm and Particle Swarm Optimization

Kohei Arai[1]*, Diva Kurnianingtyas[2], Nathan Daud[3]
Faculty of Science and Engineering, Saga University, Saga, Japan[1]
Faculty of Computer Science, Universitas Brawijaya, Malang 65154, Indonesia[2, 3]

***Abstract*—This study presents a controlled proof-of-concept comparison of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for automated daily meal-plan generation. The evaluation is deliberately restricted to the data available in the current prototype: a 24-item Japanese-style food database, one nutritional profile, one random seed (42), 100 iterations, and an equal population/swarm size of 60. Both optimizers use the same scalarized fitness function that combines nutrient-target adherence, penalties for constraint violations, and a food-diversity bonus. In the representative run, GA achieved a final fitness of 5.15 compared with 4.92 for PSO and reached its plateau by generation 14, whereas PSO reached its final value near iteration 75. The GA solution had a daily cost of ¥980, while the PSO solution had a lower daily cost of ¥650. Both solutions remained within the specified macronutrient tolerance bounds. These results reveal a configuration-specific trade-off: GA produced a higher scalar fitness and earlier convergence, whereas PSO produced a lower-cost plan and smaller deviations for several macronutrients. Because the experiment uses a single profile and a single random seed, the findings are descriptive and should not be interpreted as statistical evidence of general algorithmic superiority.**

***Keywords*—Automated meal planning; genetic algorithm; particle swarm optimization; constrained combinatorial optimization; personalized nutrition; computational intelligence**

I. Introduction

Personalized meal planning can be formulated as a computational decision problem in which nutritional adequacy, dietary restrictions, user preferences, and affordability must be considered simultaneously. Recent artificial-intelligence-based meal-planning studies have explored reinforcement learning and hybrid optimization to tailor food recommendations to individual needs and preferences [1], [2]. Population-based optimization has also been applied to nutrition- and budget-aware menu generation, including maternal nutrition planning [3].

From an optimization perspective, meal planning is a constrained combinatorial search problem because food choices are discrete and multiple criteria must be satisfied at the same time. Contemporary diet-optimization studies show that mathematical models can balance nutritional adequacy with affordability, acceptability, food waste, and other practical objectives [4], [5], [6], [7]. Population-based metaheuristics such as GA and PSO are therefore attractive when the search space is discrete or mixed, the objective is nonlinear, and feasibility is controlled through penalties or decoding rules [7], [8].

This study makes four contributions within the scope of the available prototype data. First, it formulates four-meal daily planning as a constrained combinatorial optimization problem over a 24-item food database. Second, it implements GA and PSO under matched computational settings and a common scalarized fitness function so that differences are not caused by different objective definitions. Third, it contrasts an integer-based GA representation with a continuous latent-space PSO representation that is decoded into discrete food selections. Fourth, it provides a descriptive head-to-head comparison using final fitness, convergence behavior, daily cost, and macronutrient deviations. The scope is intentionally limited to one user profile and one random seed; therefore, the study evaluates the behavior of the two implementations under a controlled configuration rather than claiming population-level or statistically generalizable superiority.

The remainder of this study is organized as follows. Section II reviews recent work on GA, PSO, and computational meal planning. Section III defines the decision variables, constraints, and fitness function. Section IV describes the system architecture, Sections V and VI present the algorithm design and implementation, Section VII specifies the experimental configuration, and Section VIII reports the descriptive results. Section IX discusses the observed trade-offs and limitations, followed by the conclusion and future research directions.

II. Research Background and Related Work

A. Genetic Algorithm (GA)

GA is an evolutionary optimization technique that improves a population of candidate solutions through selection, crossover, mutation, and survival mechanisms. Recent reviews emphasize that representation, fitness design, crossover, mutation, and diversity preservation strongly influence search behavior and convergence [7], [9], and the same body of work shows that GA and PSO performance is sensitive to control-parameter choices such as mutation/inertia settings and population size [11]. For the present meal-planning problem, a direct integer representation is appropriate because candidate solutions are constructed from discrete food identifiers grouped into meal slots.

*Corresponding author.

B. Particle Swarm Optimization (PSO)

PSO is a population-based swarm-intelligence method in which particles update their positions using their own best-known position and the best position discovered by the swarm. Although PSO is naturally defined in continuous space, it has been widely adapted to constrained, discrete, and application-specific optimization through customized encodings and decoding mechanisms [8]. In this study, each continuous particle position is interpreted as a food-selection strength and is decoded into a discrete meal plan.

$$v_i(t+1) = \omega\, v_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (1)$$

where, ω is the inertia weight controlling the exploration-exploitation balance, c_1 and c_2 are the cognitive and social acceleration coefficients, and r_1 and r_2 are uniformly distributed random scalars. For the discrete meal-selection task, the continuous position vector is not used directly as the final plan; instead, it is decoded by selecting the highest-strength eligible foods for each meal.

C. Related Work

Recent computational nutrition research has moved beyond static rule-based recommendation toward optimization and adaptive decision support. Authors in [1] used reinforcement learning to generate personalized meal plans with an emphasis on user acceptance. Authors in [2] combined PSO and simulated annealing with analytic hierarchy process weighting for personalized meal planning under multiple user-specific criteria. Authors in [3] compared several population-based optimizers, including GA and PSO, for pregnancy menu planning using nutritional requirements and food budget as optimization considerations.

Other recent diet-optimization studies demonstrate that food planning can be framed around multiple practical constraints. Authors in [4] incorporated nutrition, cost, package sizes, and food waste into a mathematical meal-planning model, while [5] optimized diet structures to improve nutritional adequacy while accounting for affordability and environmental implications. Authors in [6] further showed that nutritional requirements can materially constrain optimized healthy eating patterns. The present study is narrower in scope than these works: it does not model a population, weekly planning, sustainability, or clinical outcomes. Instead, it isolates a controlled GA-versus-PSO comparison for a four-meal daily plan using the same 24-food database, the same scalar fitness function, the same population size, and the same iteration budget. This controlled design supports implementation-level comparison, while the single-profile and single-seed scope limits statistical generalization.

III. Problem Formulation

A. Decision Variables

A daily meal plan is represented as a set of four meals $M = \{breakfast, lunch, dinner, snack\}$. Each meal $m \in M$ contains a subset of food items $Fm \subseteq F$ from the food database F, where, $|F| = 24$. Each food item $f \in F$ is characterized by its nutritional content per serving (kcal, protein, fat, carbohydrates), unit price (JPY), food category, and allergen list.

B. Constraints

The following constraints must be satisfied:

- Nutrient bounds: target caloric intake within ±10%, protein within +20%/−15%, fat within ±20%, carbohydrates within +15%/−15%.
- Allergen exclusion: no food item containing a user-specified allergen may appear in any meal.
- Dislike exclusion: explicitly disliked food IDs are permanently excluded.
- Budget constraint: the total daily cost $\sum price(f)$ must not exceed the daily budget (default: ¥1,500).
- Meal structure: each meal may contain 1–4 food items drawn from meal-appropriate categories (e.g., grains, proteins, vegetables, dairy).

C. Fitness Function

A single scalar fitness value is used to rank candidate meal plans. The objective combines nutritional adequacy, penalties for constraint violations, and a diversity reward. Accordingly, the present implementation is a scalarized multi-criteria optimization model rather than a Pareto-based multi-objective optimizer.

$$F(plan) = Snut - Cviolation + \lambda \cdot diversity(plan) \quad (2)$$

The nutrient score $Snut = \frac{\Sigma j\, wj}{(1 + devj)}$ rewards proximity to nutritional targets, where $devj = max(0, actualj - ubj, lbj - actualj)$ represents the out-of-bound deviation for nutrient j and wj is its weight; protein is weighted 1.5× in the current implementation. Constraint penalties subtract $Callergen$ = 100 for an allergen violation, $Cdislike$ = 10 for each disliked item, and 0.05 × overage for a budget violation. The diversity term is $\lambda \cdot log(1 + uniquefoods)$, with λ = 0.3. Because all components are aggregated into one fitness value, the reported fitness should be interpreted according to this specific weighting and penalty structure.

IV. System Architecture

The prototype uses a backend-centric architecture (Fig. 1). A RESTful POST /plan endpoint receives a user-profile JSON payload containing nutritional targets, dietary constraints, and the selected optimizer (GA or PSO). The optimization engine executes server-side and returns a structured meal plan, per-food nutritional information, total cost, and optimizer-trace metadata. Both optimizers access the same food database and the same fitness engine, which keeps the comparison focused on the search mechanisms rather than on different data or objective functions.

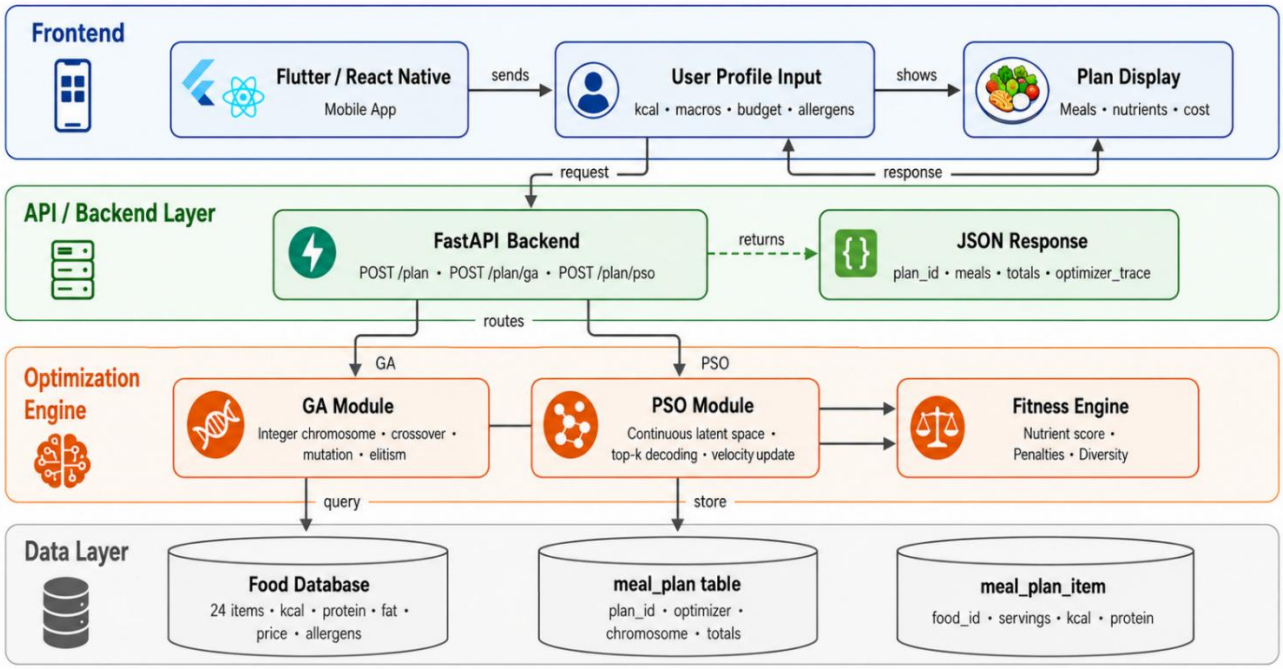


Fig. 1. System architecture of the automated daily meal-planning prototype. The frontend captures the user profile and displays the generated plan; the FastAPI backend routes requests to either the GA or PSO module; and both optimizers use the same fitness engine and data layer.

Table I summarizes the prototype components. Flutter/React Native is used for the frontend, FastAPI handles JSON-based communication, GA and PSO provide alternative optimization engines, and a lightweight SQLite/JSON food store contains the nutritional, price, category, and allergen attributes required by the fitness function.

TABLE I. SYSTEM COMPONENT OVERVIEW

Layer	Component	Responsibility
Frontend	Flutter / React Native	User profile input; plan display
API	FastAPI (Python)	POST /plan endpoint; JSON I/O
Engine	GA / PSO modules	Combinatorial meal search
Data	Food DB (SQLite/JSON)	Nutritional facts; allergen flags

V. ALGORITHM DESIGN

A. GA Encoding and Operators

Each GA chromosome is an integer-keyed meal matrix. The genome is divided into four meal slots, each containing a variable-length list (1–4) of food IDs drawn from the meal-appropriate subset of the food database. Initialization randomly samples valid foods per meal, ensuring allergen compliance. The key operators are:

1) Selection: tournament selection with k = 3 candidates, choosing the individual with the highest fitness.

2) Crossover: for each meal slot, one parent's foods are inherited with probability 0.5 (uniform crossover at the meal level).

3) Mutation ($rate\ pm = 0.15$): per meal, randomly replace a food with another valid food, insert an additional food ($if\ |meal| < 4$), or remove a food ($if\ |meal| > 1$).

4) Elitism: the top 10% of individuals are carried forward unchanged to the next generation. This rate follows a commonly used default in the GA literature rather than a value selected through tuning for the present problem, and it is one of several factors that may contribute to GA's faster observed convergence, since preserving the fittest individuals each generation directly protects incremental fitness gains from being lost to crossover or mutation. Its individual contribution relative to selection pressure and mutation rate was not isolated in this study.

B. PSO Encoding and Decoding

Each PSO particle is a real-valued vector $x \in [0,1]^N$, where $N = 24$ is the number of foods. Component xi represents the selection strength of food i. At each iteration, particle positions are decoded into discrete meal plans by selecting, for each meal m, the top- k foods ($k \in \{1,2,3\}$ sampled uniformly) whose selection strengths are highest among the meal-appropriate, allergen-free food pool. Velocity updates follow the standard inertia-weight PSO equation with $\omega = 0.7$, $c_1 = c_2 = 1.5$.

The two representations determine meal size through different mechanisms. In GA, the number of foods per meal changes only through insertion and deletion mutations acting on an already-instantiated meal. In PSO, meal size is instead re-sampled at every iteration, since k is drawn uniformly from {1, 2, 3} independently for each meal each time a particle is decoded. This introduces a source of per-iteration randomness in PSO's meal construction that has no counterpart in GA's representation. Consequently, the two optimizers are matched in population/swarm size, iteration budget, random seed, and fitness function, but not in the decoding-level stochasticity of meal length; the comparison should accordingly be read as

controlled at the level of computational budget and objective function rather than as a fully matched search-space representation.

VI. Implementation

Both algorithms are implemented in Python 3 without external optimization libraries so that the encoding, operators, decoding rules, and fitness calculations remain explicit in the prototype code. The implementation consists of approximately 300 lines organized into the following modules:

1) Food database: A list of 24 food items (Japanese-style) with nutritional profiles, prices, allergen flags, and meal-category labels.

2) Fitness engine: A shared function computing the multi-component fitness score used by both GA and PSO without modification.

3) GA module: GeneticAlgorithm class with configurable population size, number of generations, mutation rate, and elite ratio.

4) PSO module: PSO class with configurable swarm size, iteration count, inertia weight, and cognitive/social coefficients.

The algorithmic settings used for the representative comparison are shown in Table II.

TABLE II. Hyperparameter Settings used in the Representative Experiment

Parameter	GA	PSO
Population/Swarm size	60	60
Iterations/Generations	100	100
Mutation rate (pm)	0.15	-
Elite ratio	10%	-
Inertia weight (ω)	-	0.70
Cognitive Coeff. (c_1)	-	1.50
Social Coeff. (c_2)	-	1.50
Random seed	42	42

Algorithm 1 presents a representative excerpt of the GA chromosome initialization used in the prototype.

Algorithm 1: GA Chromosome Initialization

```
def _init_individual(self) -> MealPlan:
    return MealPlan(meals={mt: random_meal(mt, self.profile, random.randint(1,3))
for mt in MEAL_STRUCTURE })
```

VII. Experimental Setup

The controlled experiment uses one test profile: 2,000 kcal/day, 90 g protein, 60 g fat, 250 g carbohydrates, a ¥1,500 daily budget, and shellfish as an allergen restriction. Both implementations use seed 42, 100 iterations, and an equal population/swarm size of 60. The best-so-far fitness is recorded at each iteration. Because only one seed is available, all numerical results are descriptive single-run observations rather than estimates of expected performance.

The comparison uses final scalar fitness, plateau iteration, daily cost, and deviations from the calorie, protein, fat, and carbohydrate targets. These measures are reported directly from the available seed-42 execution and are interpreted descriptively within the scope of the prototype.

VIII. Results

Under the configured single run (seed 42), GA and PSO generated the following best daily meal plans:

GA Optimal Menu (Fitness: 5.15)

Meal Details:

Breakfast: Salmon, Tofu, Banana, Apple

Lunch: Rice, Bread, Spinach, Broccoli

Dinner: Soba Noodles, Pork Miso Soup, Chicken Breast, Cabbage

Snacks: Orange, Almonds, Rice Ball, Milk

Total: 2,045 kcal / Protein: 106g / Cost: ¥980

PSO Optimal Menu (Fitness: 4.92)

Meal Details:

Breakfast: Bread, Pork Loin, Soba Noodles

Lunch: Miso Soup, Bread

Dinner: Miso Soup, Bread, Pork Loin

Snacks: Rice Ball, Milk, Orange Total: 2,027 kcal / Protein: 94g / Cost: ¥650

Implementation characteristics relevant to interpretation of the results are summarized as follows:

- GA encoding: integer food identifiers organized by meal slot.
- PSO encoding: continuous food-selection strengths in [0,1] decoded to the highest-ranked eligible foods.
- Shared fitness function: scalar combination of nutrient adherence, constraint penalties, and food diversity.
- GA operators: tournament selection, meal-level uniform crossover, replacement/insertion/deletion mutation, and elitism.
- PSO update: inertia, personal-best attraction, and global-best attraction followed by discrete meal decoding.

For seed 42, GA achieved the higher final scalar fitness (5.15 versus 4.92). This comparison describes the configured execution only and is interpreted together with convergence behavior, daily cost, and macronutrient deviations rather than as evidence of universal algorithmic superiority.

A. Convergence Behavior

Fig. 2 shows the best-so-far fitness trace over 100 iterations for the representative run. GA starts at 3.28, reaches 5.09 by generation 10, and plateaus at 5.15 by generation 14. PSO starts at 3.67, reaches 4.88 by iteration 10, and improves to its final value of 4.92 near iteration 75, as shown in the full-resolution

curve in Fig. 2. Table III samples only four checkpoints (iterations 1, 10, 25, and 100) for brevity; it does not include a point at iteration 75, so readers should consult Fig. 2 for the exact location of PSO's late-stage increase. Relative to each algorithm's own final plateau in this run, GA required about 5.4 times fewer iterations than PSO (14 versus 75) to reach its final value. This pattern is consistent with the direct discrete GA representation and meal-level recombination implemented in the prototype and is interpreted only within the current configuration. Table III reports representative fitness values at selected iterations and complements the full convergence trace in Fig. 2. Macronutrient-target adherence is examined separately in Fig. 3.

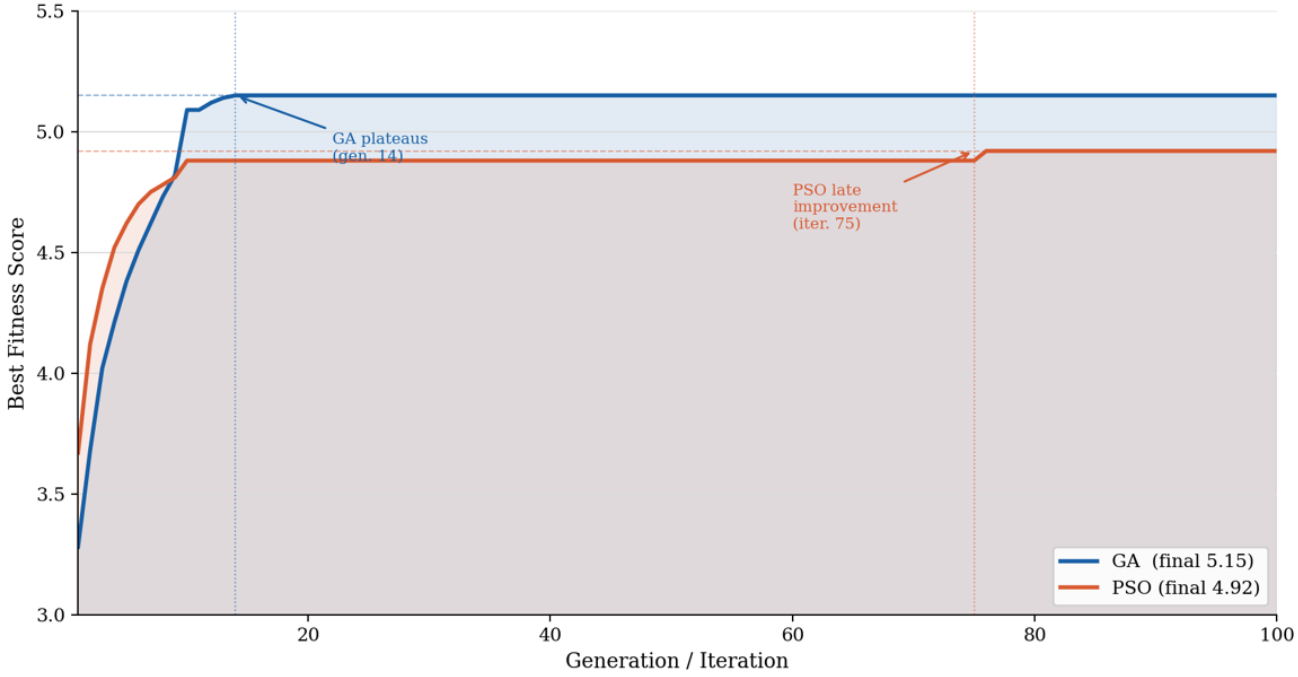


Fig. 2. Representative best-fitness convergence curves for seed 42 over 100 iterations. GA plateaus at 5.15 by generation 14, whereas PSO's fitness plateaus at 4.88 and then increases again near iteration 75, reaching its final value of 4.92. The figure is descriptive of this single run and is not an averaged convergence profile.

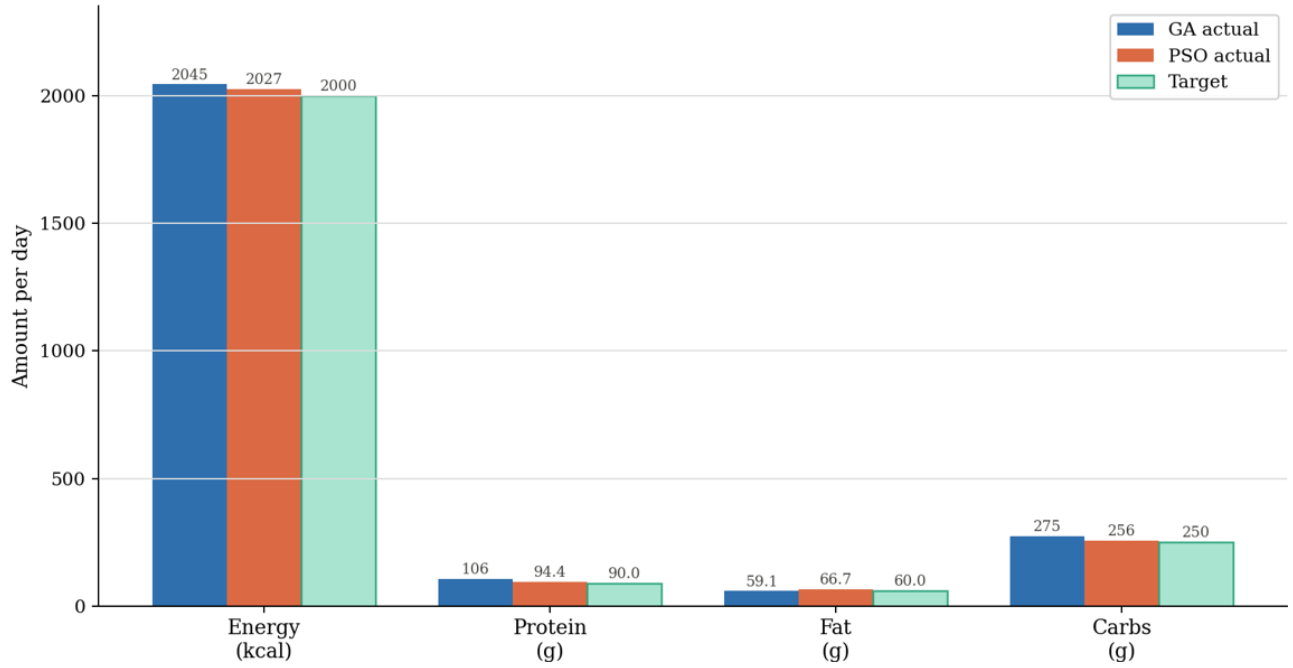


Fig. 3. Macronutrient totals for the representative GA and PSO best plans compared with the daily targets. The plotted values correspond to the single seed-42 experiment.

TABLE III. REPRESENTATIVE CONVERGENCE TRACE FOR SEED 42

Iteration	GA Fitness	PSO Fitness
1	3.28	3.67
10	5.09	4.88
25	5.15	4.88
100	5.15	4.92

These four checkpoints were chosen for brevity and do not include iteration 75, near which PSO's fitness rises from 4.88 to its final value of 4.92; this transition is visible in the full-resolution trace in Fig. 2.

B. Nutrient Accuracy

Fig. 3 compares the macronutrient totals of the best GA and PSO plans with the specified targets. The GA plan provides 2,045 kcal, 106.1 g protein, 59.1 g fat, and 275.0 g carbohydrates, corresponding to deviations of +2.3%, +17.9%, −1.5%, and +10.0%, respectively. The PSO plan provides 2,027 kcal and 94.4 g protein and, according to Table IV, shows smaller deviations than GA for energy, protein, and carbohydrates, while GA is closer to the fat target. Both plans remain within the tolerance bounds encoded in the current fitness formulation.

C. Summary

Representative final fitness and daily cost are GA = 5.15 / ¥980 and PSO = 4.92 / ¥650. Table IV consolidates the raw head-to-head metrics used in the subsequent discussion.

TABLE IV. DESCRIPTIVE HEAD-TO-HEAD COMPARISON FOR ONE PROFILE AND SEED 42

Metric	GA	PSO
Final fitness	5.15	4.92
Convergence generation	~14	~75*
Daily cost (¥)	980	650
Caloric accuracy ($\Delta kcal$)	+45 (+2.3%)	+27 (+1.4%)
Protein accuracy (Δg)	+16.1 (+17.9%)	+4.4 (+4.9%)
Fat accuracy (Δg)	−0.9 (−1.5%)	+6.7 (+11.2%)
Carbohydrate accuracy (Δg)	+25 (+10.0%)	+5.8 (+2.3%)

*PSO's convergence generation is read from the full-resolution trace in Fig. 2. Table III samples only iterations 1, 10, 25, and 100 for brevity and does not itself include a point at iteration 75.

IX. DISCUSSION

A. Advantages of GA for Meal Planning

The earlier GA plateau observed in Fig. 2 is consistent with the representation used in this prototype. Integer chromosomes directly encode discrete food selections, and meal-level crossover can preserve and recombine partial meal structures. Genetic-algorithm performance is known to depend strongly on representation, selection, crossover, mutation, and parameter settings rather than on the algorithm label alone [7], [9]. A more directly comparable meal-planning study by [3] evaluated GA together with PSO, duck swarm algorithm (DSA), and whale optimization (WO) on a 55-food maternal menu-planning problem. That study reported statistically different fitness behavior among the evaluated optimizers. The present experiment likewise shows different GA and PSO behavior in the representative run, with GA reaching a higher final fitness and an earlier plateau. However, the fitness formulations, objective directions, datasets, and experimental protocols differ across the two studies. Their numerical fitness values should therefore not be compared directly.

B. Advantages of PSO for Budget Efficiency

PSO produced a best plan costing ¥650 compared with ¥980 for GA, a reduction of approximately 33.7%, while remaining within the encoded nutritional tolerance bounds. It also achieved smaller deviations for energy, protein, and carbohydrates in the reported solution. This trade-off is relevant to recent personalized nutrition research, where cost and nutritional criteria are often optimized jointly rather than treated as a single performance dimension [2], [5]. For example, [2] showed that a hybrid PSO-Simulated Annealing approach could improve the accuracy of standard PSO when optimizing consistency in an AHP-based personalized meal-planning framework. This illustrates that PSO performance can change substantially with hybridization and problem representation. More broadly, the systematic review by [8] emphasizes that PSO behavior is sensitive to parameter settings, variants, and the balance between exploration and exploitation. Thus, the lower-cost solution obtained here should be interpreted as a configuration-specific result of the continuous latent representation, top-k decoding rule, and current fitness weights, rather than as evidence that PSO generally produces cheaper meal plans.

C. Comparison with Recent Meal-Planning Studies

The present study occupies a narrower but complementary position relative to recent AI-based meal-planning research. Amiri et al. [1] combined reinforcement learning and collaborative filtering and evaluated their approach on a large recipe-and-user-interaction setting, explicitly targeting user acceptance, preference, and nutritional adequacy. By comparison, the current prototype uses a 24-item food database and a single static nutritional profile, with no feedback loop or learned preference model. Its contribution is therefore not adaptive personalization, but a controlled implementation-level comparison of two population-based optimizers under the same food data, constraints, population size, iteration budget, and scalar fitness function.

The observed tension between GA's higher scalar fitness and earlier convergence and PSO's lower cost also reflects the broader multi-criteria nature of diet optimization. Liu et al. [5] demonstrated that diet optimization at a much larger scale requires simultaneous consideration of nutrition, affordability, acceptability, and environmental consequences. Dussiot et al. [6] separately showed that realistic nutrient requirements, particularly micronutrient constraints, can materially alter feasible optimized diets. These studies reinforce two implications for the present results. First, no single metric should be treated as sufficient evidence of overall meal-plan quality. Second, the apparent preference between GA and PSO may change when additional criteria such as micronutrients, portion sizes, acceptability, or longer planning horizons are introduced. Direct numerical benchmarking against these studies would be inappropriate because their decision variables, objectives, datasets, and evaluation protocols differ substantially from those used here.

D. Limitations

This study has several important limitations. First, the food database contains only 24 Japanese-style items, which limits the search-space diversity and does not represent a comprehensive real-world food catalog. Second, only one nutritional profile and one random seed (42) are evaluated. The study therefore reports descriptive single-run results rather than mean performance, standard deviation, or significance testing across repeated runs, and the findings should not be read as statistically generalizable. Established nonparametric methodologies for comparing stochastic optimizers across repeated runs, such as the Wilcoxon signed-rank test, are available for this purpose and are recommended for future multi-seed replication [12]. Third, portion sizes are fixed at one serving per selected item, preventing continuous portion adjustment. Fourth, the fitness function models calories and three macronutrients but not micronutrients, meal timing, clinical conditions, or longitudinal variety. Fifth, neither GA's operator settings (mutation rate, elite ratio, tournament size) nor PSO's coefficients (inertia weight, cognitive and social coefficients) were tuned or subjected to a parameter-sensitivity analysis. Both GA and PSO performance are known to depend strongly on such settings [7]–[9], [11], so the reported comparison reflects one arbitrarily chosen configuration per algorithm rather than each algorithm's best achievable performance. Sixth, the manuscript does not report the wall-clock or computational runtime of the two optimizers, even though runtime is a practically relevant dimension for comparing population-based metaheuristics and has been highlighted in prior GA–PSO comparisons [10]. These limitations define the present work as a focused proof-of-concept comparison and should be considered when interpreting the reported GA–PSO trade-offs.

X. Conclusion

This study presents a controlled proof-of-concept comparison of GA and PSO for automated four-meal daily planning using a common 24-food database, identical population/swarm size, an equal iteration budget, and the same constraint-aware scalar fitness function. Both algorithms produced meal plans that satisfied the encoded macronutrient tolerance bounds within the representative execution.

For the single profile and seed 42, GA achieved the higher final fitness (5.15 versus 4.92) and reached its final plateau earlier (generation 14 versus iteration 75, per the full-resolution trace in Fig. 2). PSO, however, produced the lower-cost plan (¥650 versus ¥980) and smaller deviations from the energy, protein, and carbohydrate targets.

Accordingly, the present evidence supports a configuration-specific trade-off rather than a universal recommendation of one optimizer. Under the current scalar fitness, GA is preferable when the emphasis is on the reported fitness value and convergence behavior, whereas the PSO solution is attractive when lower cost and closer matching of several macronutrient targets are prioritized. The conclusions are intentionally confined to the available prototype data and the representative seed-42 execution.

XI. Future Research Directions

Future research may extend the present proof-of-concept by enlarging the food database to represent a broader set of realistic food choices and by allowing serving sizes to vary rather than remaining fixed at one portion per selected item. Methodological extensions may also combine discrete food selection with continuous portion-size optimization or adopt Pareto-based multi-objective optimization when nutrient deviation, cost, and diversity are treated as separate competing objectives. The nutritional scope may be broadened from calories and macronutrients to relevant micronutrients, while longer planning horizons could support weekly or monthly menus with cross-day variety and meal-timing constraints. Further application-oriented work may additionally consider multiple nutritional profiles and domain-expert or user assessment to examine the practical usefulness of the generated meal plans.

Declaration on Generative AI

Generative AI tools were used only for figure-design assistance. The authors independently verified all scientific content, data, results, interpretations, and references and take full responsibility for the manuscript.

References

[1] M. Amiri, F. Sarani Rad, and J. Li, “Delighting palates with AI: Reinforcement learning’s triumph in crafting personalized meal plans with high user acceptance,” Nutrients, vol. 16, no. 3, p. 346, 2024, doi: 10.3390/nu16030346.

[2] F. Sarani Rad, M. Amiri, and J. Li, “Optimizing nutritional decisions: A particle swarm optimization–simulated annealing-enhanced analytic hierarchy process approach for personalized meal planning,” Nutrients, vol. 16, no. 18, p. 3117, 2024, doi: 10.3390/nu16183117.

[3] D. Kurnianingtyas, N. Daud, and K. Arai, “Automated menu planning for pregnancy based on nutrition and budget using population-based optimization method,” IAES International Journal of Artificial Intelligence, vol. 14, no. 5, pp. 3483–3492, 2025, doi: 10.11591/ijai.v14.i5.pp3483-3492.

[4] M. A. van Rooijen, J. C. Gerdessen, G. D. H. Claassen, and S. De Leeuw, “Optimizing household food waste: The impact of meal planning, package sizes, and performance indicators,” Resour. Conserv. Recycl., vol. 205, p. 107559, 2024, doi: 10.1016/j.resconrec.2024.107559.

[5] X. Liu, L. Xin, and X. Li, “The global nutrition can be greatly improved with diet optimization,” Resour. Conserv. Recycl., vol. 202, p. 107343, 2024, doi: 10.1016/j.resconrec.2023.107343.

[6] A. Dussiot et al., “Modeled healthy eating patterns are largely constrained by currently estimated requirements for bioavailable iron and zinc—a diet optimization study in French adults,” Am. J. Clin. Nutr., vol. 115, no. 3, pp. 958–969, 2022, doi: 10.1093/ajcn/nqab373.

[7] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” Multimed. Tools Appl., vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.

[8] A. G. Gad, “Particle swarm optimization algorithm and its applications: A systematic review,” Archives of Computational Methods in engineering, vol. 29, no. 5, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.

[9] R. Chaudhary, P. Verma, R. Salgotra, and A. H. Gandomi, “Five decades of genetic algorithms: A systematic and bibliometric review (1975–2025),” Archives of Computational Methods in Engineering, pp. 1–75, 2026, doi: 10.1007/s11831-025-10487-2.

[10] Ali Fozooni, Osman Kamari, Mostafa Pourtalebiyan, Masoud Gorgich, Mohammad Khalilzadeh, Amin Valizadeh, An Analysis of the Operation Factors of Three PSO-GA-ED Meta-Heuristic Search Methods for Solving a Single-Objective Optimization Problem, Comput Intell Neurosci. 2022 Oct 14;2022:2748215. doi: 10.1155/2022/2748215.

[11] Georgios Papazoglou, Pandelis Biskas, Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem, Energies 2023, 16(3), 1152; https://doi.org/10.3390/en16031152.

[12] J. Carrasco, S. García, M.M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm and Evolutionary Computation, 2020, 100665, 54, https://doi.org/10.1016/j.swevo.2020.100665.

AUTHOR'S PROFILE

Kohei Arai received BS, MS, and PhD degrees in 1972, 1974, and 1982, respectively. He was with the Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 and was with the National Space Development Agency of Japan from January 1979 to March 1990. From 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He was a professor at Saga University from 1990 to 2017. Also, he was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology from 1998 to 2000 and was a councilor of Saga University in 2002 and 2003. He was also an executive councilor for the Remote Sensing Society of Japan from 2003 to 2005. He was the Vice Chairman of the Scientific Committee and an Award Committee member of ICSU/COSPAR. He has been a Science Council of Japan Special Member (COSPAR Committee) since 2012. He is an Adjunct Professor at Nishi-Kyushu University and Kurume Institute of Technology, as well as Prishtina International University. He has written 134 books and published 779 journal papers as well as 612 conference papers. He received 66 awards, including ICSU/COSPAR Vikram Sarabhai Medal in 2016, and Science award of Minister of Ministry of Education of Japan in 2015. He is now Editor-in-Chief of IJACSA.
https://orcid.org/0009-0001-6433-1592,
https://scholargps.com/scholars/84340950120240/kohei-arai,
http://teagis.ip.is.saga-u.ac.jp/index.html

Critical Vendor-Related Failure Factors in Outsourced ICT Projects: Insights from the Malaysian Public Sector

Masittah Musatapa[1], Fazlina Mohd Ali[2], Nurhizam Safie Mohd Satar[3], Azran Ahmad[4], Surya Sumarni Hussein[5]
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia[1, 2, 3]
Department of Computing (CCI), College of Computing and Informatics,
Universiti Tenaga Malaysia, 43000, Kajang, Selangor, Malaysia[4]
Faculty of Computer & Mathematical Science, University Teknologi Mara, 40450, Shah Alam, Selangor, Malaysia[5]

***Abstract*—ICT projects in the Malaysian public sector continue to experience failure, especially in outsourced application system projects. In striving to overcome this issue, the government is leading a significant push toward the digitization of public services. In previous studies, the challenges of ICT implementation have been analyzed from the perspective of project, system, and technology management. However, the focus in these studies is typically on challenges that occur within the system. There is limited concern for the outsourcing of public services. Specifically, this study focuses on the critical factors of vendor-related failure in ICT projects in the Malaysian public sector. The study employed an interview-based approach in which eight experienced ICT practitioners involved in project implementation, procurement, governance, and vendor management participated in purposefully selected, semi-structured interviews. Data were analyzed by thematic coding of interview transcripts. Each percentage reported represents the frequency of the reference within its own HOPT-FIT domain rather than a total share across all four domains. The findings were that vendor ineffectiveness, lack of project management, problems with requirements and risk management, as well as issues of knowledge transfer and effective governance, technology, and project management were the dominant factors contributing to ICT project failure. The research further confirmed that ICT project failure is multifaceted and occurs due to the integration of challenges in the human, organizational, process, and technology domains throughout the project lifecycle. deficiencies in technology, as opposed to upstream weaknesses in human capability and process governance. This demonstrates the study's finding that human and process-related capabilities are the dominant factors contributing to the failure of outsourced ICT projects, with Technology (the biggest of these factors, technical difficulties, 75%) recording the highest percentage in a single domain. Through this study, providing the first empirical evidence on the failure of ICT projects due to vendors, I offer contributions to the body of knowledge in information systems and provide practical advice for improving the procurement of goods and services and the assessment of vendors. The results lay the foundation for the development of a vendor selection tool that aims to decrease the risk of failure of public sector projects in Malaysia and increase the chances of succeeding on the public sector digital initiatives.**

***Keywords*—ICT project; ICT project failure; project management; ICT procurement; vendor characteristics; information systems**

I. Introduction

Information and Communication Technology (ICT) drives economic growth, restructures industries, and reforms public services [1]. Today, governments use ICT application systems to offer better services, increase efficiency, build transparency, and aid evidence-based decision-making [2], [3]. The development of digital technology has made ICT infrastructures and application systems essential to deliver public services [3], [4]. Consequently, governments [5] have dedicated more resources to ICT projects to ensure that the government's digital services are effective and on par with the needs of its citizenry.

In Malaysia [6], ICT is critical for public sector transformation. Government agencies perform a myriad of ICT activities, including the development of application systems, digital service delivery projects, and initiatives for cybersecurity, systems integration, and the modernization of ICT infrastructure [6], [7]. Such initiatives are expected to improve organizational performance and remove bureaucratic bottlenecks to improve the quality and efficiency of public services [8], [9]. Thus, the successful implementation of ICT projects is vital [3] to ensure public investments produce the expected operational, organizational, and societal benefits. From an Information Systems (IS) perspective, project success is more than completing a project within time, cost, and scope constraints [10]. Instead, [11] for an ICT implementation to be successful, there has to be a congruence of the organization's objectives, business processes, and the needs of system users, and the organization's capacities [11], [12]. If vendors lack the required capabilities to support this alignment, the resulting systems will fail to provide the expected value to the organization.

Consequently, [13] argued that generic IS outcomes such as system quality and implementation outcomes, in addition to the quality and impact of services, information, and user satisfaction, are all reliant upon vendor performance. Therefore, the vendor selection process [14] should embrace an element of consideration beyond a procurement function and should reflect an IS strategic function, as it may determine the success or failure of an ICT project implementation.

Although there are advancements in digital technology and project management systems, failure of ICT projects is still

prevalent globally[15] and is still a major issue. Many publications [16], [17] have reported issues associated with ICT projects in terms of delays, budget overruns, increasing work requirements, declining quality, and seldom complete project refusals. The challenges of ICT project implementation include technological uncertainty, continually changing needs of the users, resistance of the organization, and the many kinds of stakeholders [18]. As a result, these factors ultimately lead to the failure [17] of many ICT projects despite ample planning and substantial effort. According to [17], the outcomes of ICT project failures extend beyond financial losses. Unsuccessful projects [18] tend to cause operational inefficiencies, lower user satisfaction, service disruptions, reputational damage, and missed organizational benefits [8], [14], [18]. In the public sector, these issues are particularly significant because when projects fail [19], it directly impacts how public services are provided, how well the government performs, and the level of trust from the public [11].

The Malaysian public sector faces similar issues [20]. Reports [21] and Auditor General reports [22] have identified problems with ICT project governance due to issues such as poor planning, inadequate requirements management, weak governance, limited stakeholder engagement, and deficient risk management [23]. These initiatives also face system integration issues [13] as well as vendor performance issues [8], [13]. A number of high-impact ICT projects have experienced delays in implementation, extended deadlines, increased costs, and poor functionality before they were made available for operational use [23]. The challenges of project governance and procurement processes are inadequate to manage the key success factors of these projects [20], [23].

Though a lot of work has been done looking at ICT project failure in general, a few notable gaps in the existing research remain. First, most of the existing research concentrates heavily on challenges like project management issues, organizational problems, and technological risks [16], [24], [25], while relatively fewer studies look at factors related to outsourcing as determinants of project failure [14]. Second, most research studies [14] examine project failure after implementation problems have already occurred rather than how risks start forming during the procurement and vendor selection stages. Third, few studies examine outsourced ICT project breakdowns through an IS lens within Malaysia's government sector [20], [26]. Finally, most current procurement studies focus mainly on compliance, financial assessment, and procurement efficiency, offering little theoretical grasp of how vendor characteristics [28] impact long-term ICT project results [27], [28].

Therefore, this study aims to understand how vendor traits lead to ICT project failures in Malaysia's public sector. In particular, the research intends to identify the factors related to outsourcing failures [13] and build an empirical basis to develop a better framework for selecting vendors for ICT application system projects. This study addresses gaps in the existing research and supports Malaysia's digital transformation as it also strengthens procurement theory and practice within the public sector and the IS. Understanding the main causes of failures in ICT projects is still of major concern to both researchers and practitioners [13], [29].

This section has been organized as follows: Section II describes some theories on the factors behind ICT project failures, as described in previous studies. Section III describes the methodology of this study, including a qualitative research approach and semi-structured interviews. Section IV provides empirical data and theoretical findings. Finally, Section V describes the conclusions drawn from the research.

II. THEORETICAL FINDINGS

Many governments spend an enormous amount of money on digital tools and services in efforts to develop more effective and transparent processes and communicate with their citizens more. Despite these investments, ICT project failures remain a major issue [30] in both developed and developing countries [30], [31], [32]. Various research papers have noted that numerous ICT projects end up failing, and failed implementation [32], [33] of ICT projects can be attributed to inadequate planning and initiation, insufficient budget allocation, insufficient training [34], lack of qualified personnel, poor communication to reach their goals, inefficient government practices, inadequate functionality, and unmet government expectations [35], [36]. Focusing on giving users a good experience helps increase efficiency in work. Without focused attention [48], system faults will eventually hold back overall business operations.

Additionally, studies [30] show that several factors lead to the failure of ICT projects, such as the lack of understanding of the domain agency, shortages of human resources, and poor management of stakeholder engagement [37], [38]. Referring to [39], automation projects in this sector frequently face major obstacles, leading to delays, cost overruns, and in some instances, complete failures. Moreover, the consequences of ICT project failures extend beyond financial losses [16], [17]. This leaves a gap in the systematic investigation [21] of vendor-related critical factors of failure in ICT projects. Vendor performance reliability [40] directly impacts project failure. Addressing this gap will help strengthen procurement decisions [10] and establish a foundation for successful outcomes of future government digital initiatives in [19]. Fig. 1 below shows outsourced ICT project failure factors as insights obtained from the literature.

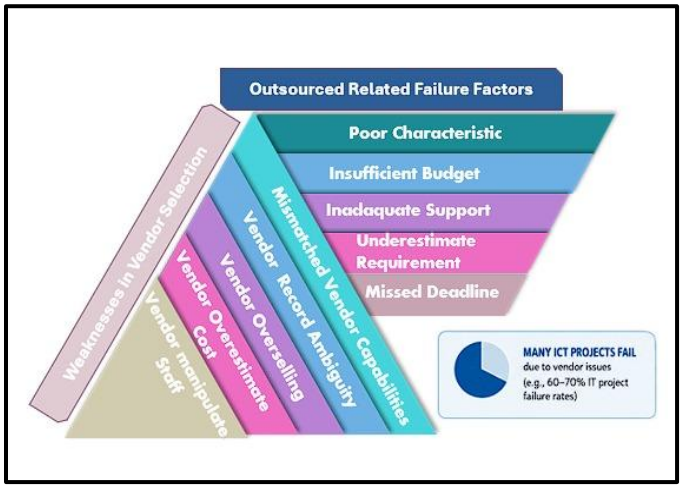


Fig. 1. Outsourced ICT project failure factors.

In Malaysia, public sector digital initiatives [41] have expanded significantly over the past decade. However, project assessments and performance audits [42] continue to report recurring implementation issues, including delayed project delivery, escalating costs, technical difficulties, integration difficulties, incomplete system functionality, and limited realization of expected benefits [13]. A lack of experience and skills related to implementation is another reason noticed for the failure of ICT projects [43]. These recurring challenges suggest that understanding the critical factors contributing to ICT project failures remains an important area of investigation, particularly in projects implemented through outsourcing arrangements.

Although numerous studies have employed various models with multiple dimensions to explore ICT projects, minimal attention has focused on the vendor's perspective. Thus, for this study, a basic human, organization, process, and technology (HOPT) have been employed as a baseline to reveal the critical failure factors in public sector ICT projects: insights from outsourced digital initiatives in Malaysia.

The increasing dependency on outsourcing has changed the implementation landscape [44] of ICT projects within the public sector. External vendors are frequently engaged to provide specialized expertise, system development services, infrastructure support, maintenance activities, and technological solutions. However, outsourcing introduces risks [44], [45] that may significantly influence project outcomes.

There are many state-of-the-art studies that explore these outsourced ICT project risks. Researchers [46] employed information, technology processes, objectives and values, staffing and skills, management systems and structures, and other resources (ITPOSMO) models to explore risks in ICT development projects. Besides, studies conducted by [47] also utilized ITPOSMO to identify gaps in e-Government implementation.

The ITPOSMO model was used by [48] and [49] in the research to evaluate the gaps between the designs realized applications, while [50] used it to identify design gaps and assess the success or failure of a statistical IS. Another opinion [48] also indicates that the ITPOSMO model is used to assess how well an IS performs in the delivery of public services and to identify areas requiring improvement to enhance employee satisfaction. Similarly, [46] reveals issues related to Enterprise Architecture (EA) implementation by breaking the factors into HOPT-FIT.

According to [51], the human, organization, and technology dimensions are essential to evaluating health information systems (HIS) implementation; [52] stated that this led to the development of a HOT-FIT model designed for HIS evaluation based on the IS success model. The HOPT-FIT model is most relevant [53] to this research due to the interrelated dimensions of human, organization, process, and technology, where each factor from the four aspects influences the others in determining the success or failure of ICT project implementation.

Based on research by [54], human involvement plays a crucial role in implementing ICT projects. The human-related dimension represents one of the most frequently reported causes of ICT project failure. The success of outsourced ICT projects depends heavily on the knowledge, skills, experience, communication abilities, and commitment of vendor personnel. Incompetency and inadequate technical expertise often limit a vendor's ability [55] to understand business requirements, develop effective solutions, and respond to implementation challenges. The organizational dimension is just as important as the human dimension, as it affects how ICT projects are steered, handled by top management, and aligned with project strategic goals. Organizations today rely on IS to carry out and streamline their business processes [56]; thus, an organization's performance relies heavily on three main factors: the quality of data in the system, user satisfaction, and the needs of the organization using the system in its everyday tasks. Poor governance structures have frequently led to project failure by slowing down decision-making processes and restricting effective project management [57].

Authors in [25] stated that process dimension is also commonly listed as key drivers of ICT project failure, as stated in. Proper execution of a project depends on clearly defined processes from initial planning of requirements, management of risks, quality checks, and ongoing monitoring. Successful implementation calls for clear processes during initial project planning, requirements tracking, risk accountability, quality checking, and project tracking; insufficiencies in these processes usually increase the uncertainty and risk of project implementation. Technology still plays a key role in shaping the results of ICT projects, especially in more complex digital transformation scenarios. Vendors must have the technical ability to build, connect, and maintain complex IS. A lack of technical expertise frequently leads to skills deficiency, implementation issues, system faults, integration problems, and reduced system performance. The dimensions and factors related to this HOPT-FIT model are depicted in Table I.

TABLE I. DIMENSION VS FAILURE FACTORS

Dimension	Failure Factors	Citation
Human	Shortage of qualified personnel, poor communication to achieve their intended objectives, incompetency of technical skills, lack of experience and skills.	[35], [33], [36], [43], [58], [34]
Organization	Human resource shortage and ineffective stakeholder involvement, delayed project delivery, financial losses	[33], [36], [59], [41], [19]
Process	Poor understanding of domain agencies, integration difficulties, ineffective government practice, inadequate functionality, unmet government expectations, poor communication, and delayed project delivery	[60], [42], [41], [13], [31], [32], [35], [32], [39]
Technology	Technical difficulties, incomplete system functionality, system defects, reduced system performance, poor system quality	[8], [42], [41], [13], [20], [61]

Table I integrates findings from various researchers and shows that the failures inherent in outsourced ICT projects are not the result of simple technical shortcomings. Instead, they result from the interrelated dimensions of people, processes, and organizations. From the human dimension, most studies point to the lack of skilled staff, poor communication, inadequate technical know-how, and lack of project experience as main factors contributing to poor collaboration, weak decision-making, and poor problem-solving in implementation. From the

organizational side, an inadequate number of staff, insufficient customer participation, late delivery of projects, and loss of funds reflect how the organization's capacity and governance contribute to a vendor's inability to fulfill projects successfully. Also, from the process side, poor understanding of government departments' needs, inefficient government processes, integration difficulties, defective systems, slow communication, and late project delivery and unmet expectations reflect that deficiencies in project management, requirements engineering, and procurement remain primary factors leading to project failure. From the technology perspective, technical difficulties, incomplete system functionality, system defects, reduced system performance, and poor system quality are frequently associated with implementation challenges that undermine system reliability and operational effectiveness.

Collectively, these studies indicate that outsourced ICT project failure results from the interaction of human capability, organizational readiness, implementation processes, and technological quality. However, most existing studies examine these factors during project implementation or post-implementation, while limited attention has been given to integrating them as vendor-related evaluation criteria during the procurement and tenderer selection stage. This gap provides the theoretical motivation for the present study to investigate critical vendor-related failure factors and to support the development of a comprehensive Tenderer Selection Specification Framework for outsourced ICT projects in the Malaysian public sector.

III. METHODOLOGY

This study adopted a qualitative research approach to obtain in-depth information on critical ICT project failure factors. This approach was conducted [24] to ensure certainty through interviews with experts in ICT project management Malaysia public sector [12], [24]. This method is appropriate to examine experts' opinions and suggestions on current failure factors that may lead to the risk of application system development failure [13].

To improve content validity, the interview protocol [62] was developed based on the collection of factors and domains through a comprehensive review and found in the related works [63], [64], [65] and was reviewed by academic experts and experienced practitioners before preliminary data collection to refine the instrument and obtain useful suggestions [67]. Feedback gathered [66] in the review process was used to make sure clarity, readability, and relevance in the final interview tool.

Semi-structured interviews [67] were chosen as the main data collection method because they allow participants to describe comprehensive experiences, views, and ideas about ICT project failures. This method helps to elicit detailed expert feedback [68] from their experiences linked to vendor behavior and project results.

The research was conducted in relation to ICT application system projects implemented in Malaysian public sector organizations [13], [69]. The public sector was chosen because government agencies keep encountering implementation problems [19], [70] despite major investments in digital transformation initiatives. Moreover, most ICT application system projects in the public sector are usually carried out through outsourcing agreements, so vendor performance is a key factor in whether the project succeeds or fails [71], [72], [73].

A purposive sampling approach [65] was used to find participants with a lot of experience in ICT project implementation, procurement management, project governance, and vendor management activities. Eight experts were selected according to predetermined inclusion criteria. First, participants needed ample professional experience in ICT project management or ICT application system implementation [74], [75]. Second, participants needed direct involvement in outsourced [13] ICT projects in the public sector. Third, participants needed hands-on experience with project planning, procurement, vendor assessment, project governance, system development, or project monitoring. Finally, participants needed to occupy more senior roles so they could offer a strategic and operational viewpoint on the failures of ICT projects [12], [65], [76].

The selected experts represented diverse professional backgrounds, including deputy ICT directors, project managers, leading PMO and PMO teams (project monitoring office), procurement officer [7]. This diversity ensured comprehensive coverage of technical, managerial, organizational, and procurement perspectives relevant to vendor-related project failures [74]. Table II below shows the expert selection criteria applied in the Malaysian public sector.

Data analysis was performed using thematic content analysis. Interview transcripts were transcribed verbatim and coded manually, with each mention of a failure factor tagged to its corresponding HOPT-FIT dimension (Human, Organization, Process, or Technology). The percentage reported for each factor represents the proportion of coded references to that factor relative to the total coded references within its own domain and therefore reflects the relative salience experts placed on that factor within each dimension rather than its share of the overall interview data across all four domains combined. To strengthen the credibility and consistency of the coding process, the coding scheme and factor categorization were independently reviewed by two academic experts, and disagreements were resolved through discussion until consensus was reached. Given the purposive and highly experienced nature of the eight participants (15–25 years of relevant experience across diverse public-sector domains), data collection continued until no new failure factors emerged from successive interviews, indicating that thematic saturation had been reached within the scope of this exploratory study. Nonetheless, the relatively small, purposively sampled group of participants is acknowledged as a limitation of this study, and this is addressed further in Section V.

As shown in Table II, the interview sessions were conducted based on the theoretical findings established in Section II, which provided the basis for identifying research questions and confirmed that this research topic is suitable within the information system field. Table II presents the expert selection criteria adopted to ensure the credibility and relevance of the qualitative findings. Eight experts (EX1–EX8) were selected based on a set of predefined criteria, including at least 15 years of professional experience in ICT project management, system development, procurement, or ICT governance. All

selected experts had considerable involvement in Malaysian public sector ICT projects and substantial experience in outsourcing evaluation, contract management, procurement processes, or the implementation of outsourced ICT projects. The selected experts also held strategic, managerial, or technical decision-making positions and had a deep understanding of the challenges related to the implementation of ICT application systems and IT projects. Participants were chosen to reflect different areas of the public sector, spanning digital services, enforcement, public security, public order, health management, and natural resources. These participants had between 15 and 25 years of professional experience, allowing them to provide valuable, authentic, and situational explanations of the critical failures experienced on outsourced public sector ICT projects in Malaysia.

TABLE II. EXPERT SELECTION CRITERIA

Experts (EX)	Experience	Position	Agency (Domain)
EX1	25 years	Deputy Director, Project Management Unit	Digital services
EX2	21 years	Deputy Director, IT Department	Enforcement
EX3	20 years	Chief Assistant Director, IT Department	Public security
EX4	20 years	Chief Assistant Director, IT Department	Public order
EX5	18 years	Assistant Director, IT Department	Health management
EX6	17 years	Assistant Director, IT Department	Enforcement
EX7	15 years	Assistant Director, IT Department	Natural resources
EX8	17 years	Deputy Director, Project Management Unit	Digital services

IV. EMPIRICAL FINDINGS

While Section II synthesized failure factors identified in prior literature to build the theoretical foundation of this study, this section presents the empirical findings derived directly from the eight expert interviews conducted for this research. The analysis of data obtained from the expert interviews revealed critical failure factors that affect the outcomes of outsourced ICT projects in the Malaysian public sector, and these empirical findings are mapped back to the theoretical findings from the literature review discussed earlier. The critical failure factors were categorized into four main domains based on the HOPT-FIT model, namely Human, Organization, Process, and Technology (HOPT).

The analysis showed that outsourced ICT projects in the public sector in Malaysia have complex failures due to the interconnectedness of different factors. The first of these is the Human domain. Insufficient communication, lack of skills, experience, and competency, as well as a lack of specialized skills and knowledge, have all been factors in project failures. The Organization domain noted issues of capacity, particularly around challenges with human resources, stakeholder inclusion, and financial shortages resulting in delayed project delivery. Among the challenges in the Process domain, the study noted poor understanding of government agency requirements, insufficient government practices, difficulties with integration, insufficient system functionality, and challenges with unmet government expectations. Technology domain issues included incomplete systems and technical problems. These findings suggest that all ICT project failures have to do with people, their organization, and the process of implementation, and the capabilities of technology, giving a full account of user-related failure factors in outsourced public sector ICT projects. The following is a comprehensive breakdown for each domain.

A. Human Domain

According to all eight experts, communication breakdown and insufficient project experience are the biggest issues in the Human domain, each contributing approximately 30% of the coded references within this domain. The results show that vendor-to-government agency communication is one of the primary causes of miscommunication during the business requirements clarification process. Poor communication causes delays, misaligns stakeholders, and results in ambiguous solutions to project issues. Similarly, project experience on the vendor's side is just as important; without experience, a vendor cannot anticipate risks during implementation, manage public sector requirements, or handle the fluctuation of project requirements. Other causal factors (approximately 15%) included incompetence, reflected in a lack of professional knowledge and poor problem-solving. Performance-related issues (approximately 13.5%) included poor commitment and inconsistency in the delivery of assigned projects. Inadequate technical expertise (approximately 11.5%) affects a vendor's ability to provide reliable and sustainable ICT solutions. Collectively, these findings show that human capacity remains one of the main obstacles to project success and vendor performance. Fig. 2 illustrates the factors identified by all experts under the Human domain.

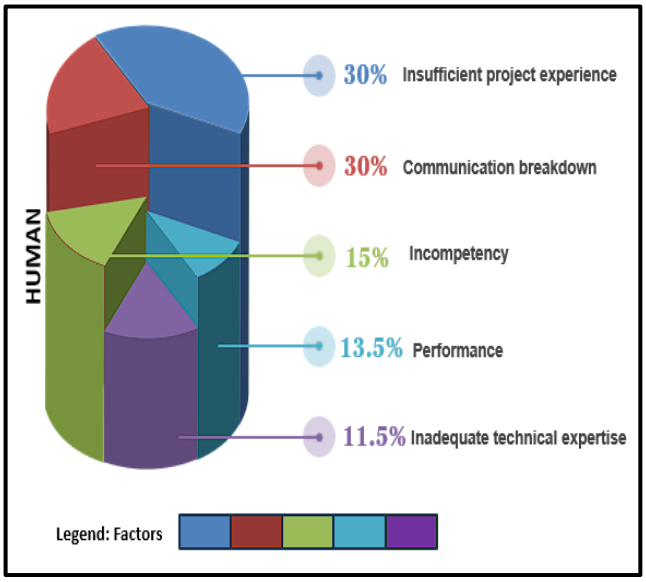


Fig. 2. Human factors related.

B. Organization Domain

According to the experts, 50% of participants identified human resource issues as the biggest organizational constraint, while the remaining 50% identified financial breakdown as the biggest constraint. Insufficient human resources among vendors means there are not enough skilled resources to carry

out the project, resulting in an overburdened workforce, reduced productivity, and delays in project activities. Financial breakdown means that a vendor lacks the financial reserves to support the project, retain skilled employees, or cover financial gaps in the required technologies and sustain the project. These constraints adversely affect the operational resilience of a vendor and its ability to deliver contracted outcomes as expected. Fig. 3 indicates the factors selected by the experts.

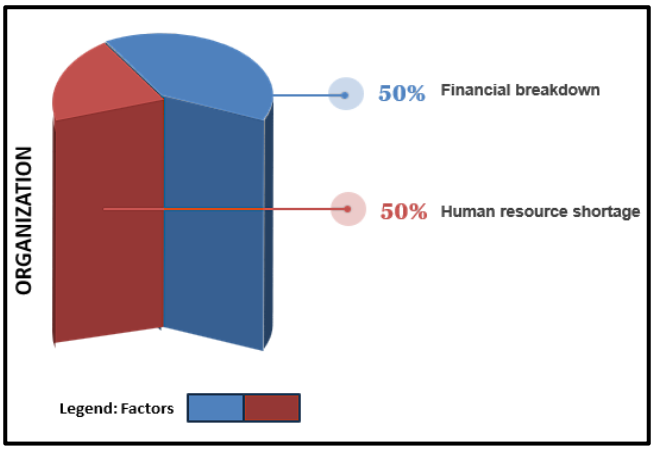


Fig. 3. Organization factors related.

C. Process Domain

The Process domain indicated that communication breakdown continued to be the most important process factor, at 40% of coded references, indicating that poor communication affects not only people but the project management process itself. Poor communication leads to issues in identifying requirements, engaging stakeholders, managing change, and coordinating the project. Additionally, long project completion times (20%), deficient government operating practices (20%), and poor understanding of government agency requirements (20%) show that issues in planning and governance, coordination of procurement, and management of requirements continue to negatively affect the effectiveness of project implementation. The results in Fig. 4 show that ineffective project processes lead to increasing risks in project cost, time, and quality.

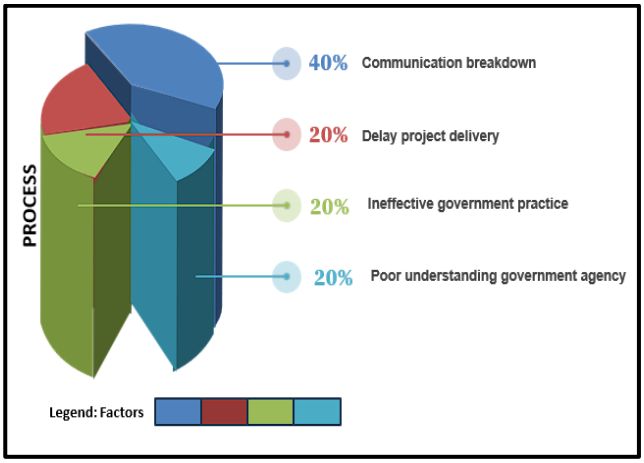


Fig. 4. Process factors related.

D. Technology Domain

Technology domain experts unanimously agreed that the biggest technological challenge for outsourced ICT projects is technical difficulties, accounting for 75% of coded references within this domain. This challenge is characterized by the integration of multiple systems, compatibility issues between software components, and insufficient infrastructure, along with the complexity and implementation challenges of the systems that lead to delays and compromise system effectiveness. Second, poor system quality (13.5%) reflects weaknesses in system functionality, reliability, and performance. Although Technology issues were reported less frequently overall than Human and Organizational issues, the experts indicated that many of the technological challenges are caused by deficiencies in the planning and governance of communication and project execution, rather than by the technology itself — reinforcing that the Human and Process domains are the primary root causes underlying downstream technical shortcomings. Fig. 5 illustrates the Technology domain factors identified by the experts.

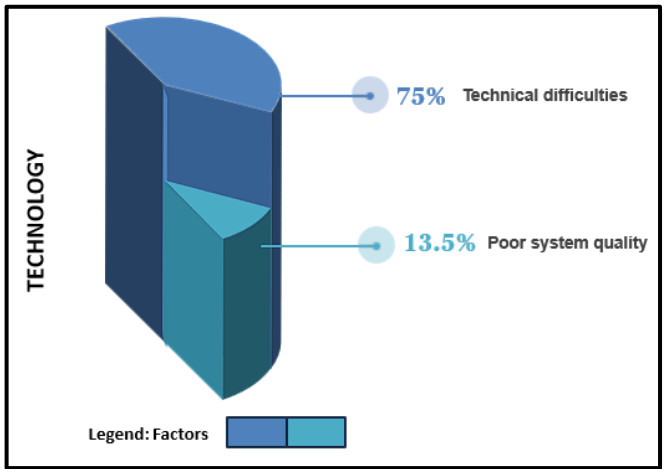


Fig. 5. Technology factors related.

The findings show how complex and interrelated the critical failure factors can be in outsourced public sector ICT projects. Weaknesses in one domain can easily affect the other domains and adversely affect a project by causing delays, budget overruns, low-quality systems, user dissatisfaction, and unmet organizational goals. These findings support the notion that for a vendor to improve its capability, there should be an equally strong focus on all elements within the human domain, the organization, the process, and the level of technical preparedness demanded across the full spectrum of the ICT project lifecycle.

The critical failure factors identified in this study provide insight into the challenges associated with the implementation of outsourced ICT projects, despite the large public commitment to digital transformation endeavors. Communication was the most critical factor across the Human and Process domains, highlighting the negative effects that ineffective collaboration between vendors and government agencies has had, and continues to have, on planning, requirements management, decision-making, and the ultimate implementation of the project. Restrictions facing organizations, primarily a lack of financial and human resources, increase the challenges vendors

face in delivering projects. Inadequate planning and execution along the governance dimension of the project are typically the root causes of malfunctions, as opposed to deficiencies in technology itself. The cumulative effect of these problems leads to delays in project completion and consequent cost overruns, poor-quality systems and services, and increased user dissatisfaction, as the organization's intended benefits are not achieved. This supports the claim that ICT failure is a multidimensional issue, and that evaluating each vendor component — people, organizational capacity, implementation process, and technical skill — is crucial. Using empirical evidence from practitioners, this study describes the challenges of outsourcing public sector digital projects and guides adjusting procurement, vendor evaluation, and project management frameworks for government services to support the success of future ICT undertakings.

V. Conclusion

This study on outsourced ICT projects in the Malaysian public sector focused on the primary vendor-related failure factors. From the interviews with ICT specialists, it was found that project failure may be attributed to a combination of human, organizational, process, and technology factors, in the absence of adequate synthesis of vendor capability across all four domains. It was found that the way a vendor synthesizes its capabilities will ultimately affect project outcomes. From an active public sector perspective, a lack of adequate communication, a lack of competency and organizational readiness, inadequate capacity for project execution, and a lack of adequate technical capabilities negatively affect digital strategies. These factors can result in poor system quality, user dissatisfaction, cost overruns, delayed project completion, and an inability to achieve intended outcomes and benefits.

This study adds value to the existing literature on ICT project failure by focusing on outsourced digital initiatives in Malaysia and will be of interest to academia. It will also be of interest to the public sector in proposing a more complete assessment of vendors and in extending risk management approaches across the project lifecycle. This study focuses on the failure of outsourced ICT projects and aims to improve procurement and project management practice. It should be noted that the findings are based on a purposively sampled group of eight experts within the Malaysian public sector, and while thematic saturation was observed within this group, further research involving a greater variety of stakeholders across the public sector is necessary to validate and extend these findings, improve strategies for mitigating the risk of ICT project failure, and enhance the success of future digital transformation initiatives.

References

[1] M. A. Kurniawati, “Analysis of the impact of information communication technology on economic growth: empirical evidence from Asian countries,” Journal of Asian Business and Economic Studies, vol. 29, no. 1, pp. 2–18, Feb. 2022, doi: 10.1108/JABES-07-2020-0082.

[2] J. Ilmiah, E. & Erudisi, U. Nurjannah, E. Hasymi, and R. E. Putera, “JIEE: Information Technology-Based Public Service Innovation Management Strategy: A Case Study of Mobile Application Usage in Government Agencies,” vol. 05, no. 1, pp. 33–49, 2025.

[3] C. Komba, “Analyzing the impact of information and communication technology (ICT) on good governance in Tanzania,” African Journal of Social Issues, vol. 7, no. 1, pp. 396–414, Mar. 2024, doi: 10.4314/ajosi.v7i1.25.

[4] A. Gonçalves, J. Varajão, P. Moura Oliveira, and I. Moura, “Success Factors for Public Sector Information Systems Projects,” Digital Government: Research and Practice, vol. 6, no. 4, Dec. 2025, doi: 10.1145/3761819.

[5] J. J. P. Latupeirissa, N. L. Y. Dewi, I. K. R. Prayana, M. B. Srikandi, S. A. Ramadiansyah, and I. B. G. A. Y. Pramana, “Transforming Public Service Delivery: A Comprehensive Review of Digitization Initiatives,” Apr. 01, 2024, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/su16072818.

[6] H. Sharma, “Optimizing government outsourcing: A performance management framework for enhancing public service delivery,” J. Public Policy, 2025, doi: 10.1017/S0143814X24000205.

[7] M. Ayat, M. Imran, A. Ullah, and C. W. Kang, “Current trends analysis and prioritization of success factors: a systematic literature review of ICT projects,” Apr. 08, 2021, Emerald Group Holdings Ltd. doi: 10.1108/IJMPB-02-2020-0075.

[8] M. S. A. M. Abdullah, A. N. Aizuddin, and M. R. Abdul Manaf, “The Impact of Provider’s Quality of Information System on User Satisfaction and Perceived Net Benefits in Malaysian Public Hospitals,” Malaysian Journal of Medical Sciences, vol. 32, no. 2, pp. 108–121, 2025, doi: 10.21315/mjms-12-2024-952.

[9] A. A. Abdullah, Z. Abdul-Samad, H. Abdul-Rahman, and H. Salleh, “Project Management Standards, Guides and Methods: A Critical Overview,” 2021.

[10] N. Ngatman, H. Alderei, H. Musa, S. Zayed Road, A. Muroor, and A. Dhabi UAE, “The Challenges faced by Public Sector Governance in implementing e-Procurement System: A Case Study of The Department of Education, Malacca,” 2020.

[11] Dr. R. Singh, “Technology-Enabled Public Service Delivery: Measuring the Impact of E-Governance on Administrative Efficiency,” International Journal of Integrated Research and Development, vol. 1, no. 1, 2025, doi: 10.63665/IJIRD.v1.i1.3.

[12] N. A. Zaid and M. Mohd Yusof, “An Evaluation of a Training Management System Adoption in A Public Sector: A Case Study,” Asia-Pacific Journal of Information Technology and Multimedia, vol. 11, no. 02, pp. 18–39, Dec. 2022, doi: 10.17576/apjitm-2022-1102-02.

[13] Majeed Zangana, “Critical Factors for Information System Project Implementation in Malaysia,” Journal of Environmental Science, Computer Science And Engineering & Technology, vol. 10, no. 1, Feb. 2021, doi: 10.24214/jecet.b.10.1.01826.

[14] P. Paleri et al., “ICBM-SBE Editorial Research Articles Rethinking Human Resource Management-Capitalizing Human: The Next Level Global Talent Management Strategies for Organizational Success in a VUCA World-A Conceptual Framework Critical Failure Factors (CFFs) of IT Projects Modelling of Dividend Determinant in Emerging Economy: An Empirical Evidence from Indian Steel Industry Diversifi cation Strategy and its Eff ects on Profi tability: A Comparative Study Critical Failure Factors (CFFs) of IT Projects,” 2016.

[15] J. Xu, “An accurate management method of public services based on big data and cloud computing,” Journal of Cloud Computing, vol. 12, no. 1, Dec. 2023, doi: 10.1186/s13677-023-00456-0.

[16] V. Swarnakar, A. R. Singh, and A. K. Tiwari, “Evaluating the effect of critical failure factors associated with sustainable Lean Six Sigma framework implementation in healthcare organization,” International Journal of Quality and Reliability Management, vol. 38, no. 5, pp. 1149–1177, Apr. 2021, doi: 10.1108/IJQRM-07-2020-0243.

[17] J. Schmidt, “Mitigating risk of failure in information technology projects: Causes and mechanisms,” Project Leadership and Society, vol. 4, Dec. 2023, doi: 10.1016/j.plas.2023.100097.

[18] B. Prayatna, M. Ihsan Suzeta, and H. Clausa, “Factors of IT Project Failure: A Literature Review Kata kunci: IT Project Failure, Failure Factors in IT Projects, Project Management Challenges,” Technologia Journal: Jurnal Informatika E-ISSN, vol. 2, no. 1, pp. 3046–9163, 2025, doi: 10.62872/jmxyt167.

[19] S. M. Mohamad Azmi and S. Ismail, “Weaknesses of Malaysian public procurement: a review of auditor general’s reports,” Journal of Financial

Reporting and Accounting, vol. 21, no. 5, pp. 999–1020, Nov. 2023, doi: 10.1108/JFRA-05-2021-0132.

[20] S. J, N. A, and P. S. ASA, "Public Sector Accountability – Evidence from the Auditor General's Reports," 2017.

[21] B. Watjatrakul and V. Vatanapitukpong, "Outsourced ERP system implementation success: The effects of client and vendor competences and their moderating roles," International Journal of Information Systems and Project Management, vol. 14, no. 1, 2026, doi: 10.12821/ijispm140104.

[22] J. D. N. JDN, "PPrISA 2.0 Versi Beta - Disember 2024," 2024.

[23] J. Pereira, J. Varajão, and N. Takagi, "Evaluation of Information Systems Project Success–Insights from Practitioners," Information Systems Management, vol. 39, no. 2, pp. 138–155, 2022, doi: 10.1080/10580530.2021.1887982.

[24] N. Wahida, M. Aron, F. Azli, M. Rahim, and N. Mardhiyah Aziz, "Analyzing The Issues Of E-Tendering Implementation In Public-Private Partnership Practise: A Preliminary Studi In Malaysia," 2024.

[25] R. Shabir, "Evaluation of the Factors behind the Failure of Project Management Practices," Journal of Development and Social Sciences, vol. 4, no. III, Sep. 2023, doi: 10.47205/jdss.2023(4-iii)47.

[26] A. Pereira-Vale, D. Perovich, and S. F. Ochoa, "Understanding the Pre-Contract Process of Small Software Projects," in Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, Knowledge Systems Institute Graduate School, 2023, pp. 146–151. doi: 10.18293/SEKE23-016.

[27] I. Jebreen and E. Al -Qbelat, "A Systematic Snapshot of Software Outsourcing Challenges," Technology, Engineering & Mathematics (EPSTEM), vol. 22, 2023, [Online]. Available: http://www.sciencedirect.com

[28] V. Wadhwa and A. R. Ravindran, "Vendor selection in outsourcing," Comput. Oper. Res., vol. 34, no. 12, pp. 3725–3737, Dec. 2007, doi: 10.1016/j.cor.2006.01.009.

[29] P. Eckersley and L. Ferry, "Public service outsourcing: the implications of 'known unknowns' and 'unknown unknowns' for accountability and policy-making," Public Money and Management, vol. 40, no. 1, pp. 72–80, Jan. 2020, doi: 10.1080/09540962.2019.1660096.

[30] D. Y. Liana, M. M. Y. Mazana, and M. Hamisi, "Factors Leading the Failure of ICT Project Management in the Public Sectors in Tanzania," European Journal of Theoretical and Applied Sciences, vol. 1, no. 4, pp. 788–798, Jul. 2023, doi: 10.59324/ejtas.2023.1(4).72.

[31] Rehan Syed and R. Eden, "Public Sector Digital Transformation Barriers: A Developing Country Experience," 2023.

[32] Y. Yao, "Digital Government Information Platform Construction: Technology, Challenges and Prospects," International Journal of Social Sciences and Public Administration, vol. 2, no. 3, pp. 48–56, Apr. 2024, doi: 10.62051/ijsspa.v2n3.06.

[33] D. F. Msilamgunda and R. O. Casmir, "The Delay and Failure of ICT Related Projects: Empirical Impediments in Ministries, Departments and Agencies in Tanzania," European Journal of Theoretical and Applied Sciences, vol. 2, no. 4, pp. 787–797, Jul. 2024, doi: 10.59324/ejtas.2024.2(4).67.

[34] M. M. Yusof, T. Takeda, Y. Shimai, N. Mihara, and Y. Matsumura, "Evaluating health information systems-related errors using the human, organization, process, technology-fit (HOPT-fit) framework," Health Informatics J., vol. 30, no. 2, Apr. 2024, doi: 10.1177/14604582241252763.

[35] S. K. Arora, "Project Failure: A Bad Communication (Case Study)," International Journal of Management and Humanities, vol. 9, no. 5, pp. 5–7, Jan. 2023, doi: 10.35940/ijmh.E1553.019523.

[36] M. Irfan et al., "Role of project planning and project manager competencies on public sector project success," Sustainability (Switzerland), vol. 13, no. 3, pp. 1–19, Feb. 2021, doi: 10.3390/su13031421.

[37] K. V. Masiba and M. S. Xegwana, "Leadership And Governance Factors In Project Failures: A Focus On Cost And Budget Management," Journal of Public Administration, Finance and Law, vol. 32, pp. 283–294, 2024, doi: 10.47743/jopafl-2024-32-21.

[38] N. Nzuza, "Success and failure of ICT for economic development (ICT4EcD) projects in the Global South: Scoping review of underlying factors Ingrid Siebörger," The African Journal of Information and Communication (AJIC) Issue, vol. 36, no. 1, 2025, doi: 10.23962/ajic.i36.

[39] M. A. T. Mohd Shukry, N. A. Muhammad, N. Panjaitan, and N. Muhammad, "Critical Failure Factors for Factory Automation Project in Medical Device Assembly," Semarak International Journal of Applied Sciences and Engineering Technology, vol. 5, no. 1, pp. 36–54, Apr. 2025, doi: 10.37934/sijaset.5.1.3654.

[40] M. S. L. D. Viduranga and S. Divakara, "Pre-Sales Pitfalls Causing Failure of IT Projects: An Exploration of Challenges Faced by Project Managers of an ICT Service Firm," Sri Lanka Journal of Management Studies, vol. 7, no. 1, pp. 36–61, Sep. 2025, doi: 10.4038/sljms.v7i1.170.

[41] P. P. Kulandaisamy, Z. Abdul-Samad, H. Salleh, A. C. Kasim, H. H. Basri, and M. B. Abdullah, "Critical Success Factors of ICT Projects in Malaysia," 2025. [Online]. Available: https://www.jisem-journal.com/

[42] Z. Abdul-Samad and P. P. Kulandaisamy, "Cost Management for Information and Communication Technology Projects," Journal of Engineering, Project, and Production Management, vol. 12, no. 2, pp. 166–178, May 2022, doi: 10.32738/JEPPM-2022-0015.

[43] R. Secchi and A. Camuffo, "Mitigating the risk of failure in lean banking implementation: the role of knowledge codification," Production Planning and Control, vol. 32, no. 12, pp. 1036–1048, 2021, doi: 10.1080/09537287.2020.1784482.

[44] V. S. Balasubramaniam, M. Mohana, and K. Dandu, "Enhancing Vendor Management for Successful IT Project Delivery," Peer Reviewed & Refereed Journal | ISSN, vol. 1, p. 370, 2024, doi: 10.36676/mdmp.v1.i.

[45] T. Huang, A. Yokota, and K. I. Suzuki, "An Empirical Study of Risk Management Approaches in Information Systems Development Projects from the Vendor's Perspective: Evidence from Japan," in 2025 14th International Conference on Software and Computer Applications, ICSCA 2025, Association for Computing Machinery, Inc, Sep. 2025, pp. 47–54. doi: 10.1145/3731806.3731837.

[46] H. Sallehudin et al., "Modelling the Enterprise Architecture Implementation in the Public Sector using HOT-Fit Framework," 2019. [Online]. Available: www.ijacsa.thesai.org

[47] D. Gusman and Y. Y. Kusuma, "GAP Analysis of E-Government Implementation In Indonesia," Jurnal Teknik Industri Terintegrasi, vol. 6, no. 1, pp. 209–215, Apr. 2023, doi: 10.31004/jutin.v6i1.13483.

[48] K. Ramanda, R. Awaludin, D. Puspitasari, and E. D. Sikumbang, "Model ITPOSMO untuk Evaluasi Keberhasilan Aplikasi Bravo," Jurnal SAINTEKOM, vol. 15, no. 1, pp. 30–41, Mar. 2025, doi: 10.33020/saintekom.v15i1.754.

[49] C. Chigwedere, "An Assessment of the Consistency Gap in Developing Countries: An Assessment of the Consistency Gap in Developing Countries: A Case of Zimbabwe's ICT Regulatory Framework A Case of Zimbabwe's ICT Regulatory Framework," 2023. [Online]. Available: https://digitalcommons.kennesaw.edu/acist

[50] A. Arief, I. H. A. Wahab, and D. I. Sensuse, "Model ITPOSMO untuk Evaluasi E-Government: Tinjauan Studi Kasus," Jurnal Ilmiah FIFO, vol. 13, no. 2, p. 106, Dec. 2021, doi: 10.22441/fifo.2021.v13i2.001.

[51] J. Xu and W. Lu, "Developing a human-organization-technology fit model for information technology adoption in organizations," Technol. Soc., vol. 70, Aug. 2022, doi: 10.1016/j.techsoc.2022.102010.

[52] E. E. Sala and A. P. Subriadi, "Hot-Fit Model to Measure the Effectiveness and Efficiency of Information System in Public Sector," The Winners, vol. 23, no. 2, pp. 131–141, May 2023, doi: 10.21512/tw.v23i2.7423.

[53] W. P. Hapsari, U. A. Labib, H. Haryanto, and W. Safitri, "A Literature Review of Human, Organization, Technology (HOT)-Fit Evaluation Model," 2021.

[54] N. Feng, Y. Chen, H. Feng, D. Li, and M. Li, "To outsource or not: The impact of information leakage risk on information security strategy," Information and Management, vol. 57, no. 5, Jul. 2020, doi: 10.1016/j.im.2019.103215.

[55] L. Barros, C. Tam, and J. Varajão, "Agile software development projects–Unveiling the human-related critical success factors," Inf. Softw. Technol., vol. 170, Jun. 2024, doi: 10.1016/j.infsof.2024.107432.

[56] M. R. H. Roslan, K. Ahmad, and M. M. Ayyash, "Factors Influencing Information Systems Quality From The System Developers Perspective," Asia-Pacific Journal of Information Technology and Multimedia, vol. 09, no. 01, pp. 82–93, Jun. 2020, doi: 10.17576/apjitm-2020-0901-07.

[57] J. Westenberger, K. Schuler, and D. Schlegel, "Failure of AI projects: Understanding the critical factors," in Procedia Computer Science, Elsevier B.V., 2021, pp. 69–76. doi: 10.1016/j.procs.2021.11.074.

[58] S. Chatterjee, R. Chaudhuri, M. Mariani, and S. Fosso Wamba, "The consequences of innovation failure: An innovation capabilities and dynamic capabilities perspective," Technovation, vol. 128, Dec. 2023, doi: 10.1016/j.technovation.2023.102858.

[59] A. Sarina, A. Zazili, A. Abu, S. A. Basri, E. Nordin, and W. S. Hussain, "Evaluating Internal Control Mechanisms in Malaysian Public Sector Initiatives: Insights from the Public Accounts Committee (PAC) Reports," 2024.

[60] S. Azhar, "The Causes Of Project Delay In The Ministry Of Defence, Malaysia," 2019.

[61] C. E. Moe and T. Päivärinta, "Challenges In Information Systems Procurement in the Public Sector," 2013. [Online]. Available: www.ejeg.com

[62] M. I. Ikhsan, N. Nion, R. S. Salleh, L. J. Matah, K. M. A. A. Ku Yusof, and S. Kamaruddin, "'Jurist': A Framework for the Creation of Interview Protocols in Qualitative Legal Research," 2026, doi: 10.55573/JISED.118542.

[63] E. Malvaceda-Espinoza and G. Flores-Pereyra, "Methodological Aspects in the Construction of the Protocol in Semi-Structured Interviews," Qualitative Report, vol. 30, no. 12, pp. 4879–4890, Dec. 2025, doi: 10.46743/2160-3715/2025.6998.

[64] D. C. Coker and A. Akande, "Preparation, Development, and Refinement of the Interview Protocol in Qualitative Research," Qualitative Report, vol. 30, no. 8, pp. 4205–4235, 2025, doi: 10.46743/2160-3715/2025.7964.

[65] R. B. Bouncken, W. Czakon, and F. Schmitt, "Purposeful sampling and saturation in qualitative research methodologies: recommendations and review," Review of Managerial Science, Feb. 2025, doi: 10.1007/s11846-025-00881-2.

[66] L. O. Akpan, "Conducting Interviews Using An Interview Schedule in Qualitative Educational Research," Mau International Journal Of Educational Foundations, vol. 2, no. 2, pp. 198–211, Jun. 2026, doi: 10.64290/mauijef.v2i2.72.

[67] M. Padilla, M. Gutierrez, and J. Fagan, "Using Semistructured Telephone Interviews to Collect Qualitative Data From People With HIV Who Are Not in Medical Care: Implementation Study," JMIR Res. Protoc., vol. 11, no. 11, Nov. 2022, doi: 10.2196/40041.

[68] B. Yayın et al., "A Qualitative Research on Being an Author, Reviewer and Editor in the Scholarly Publication Process," 2024.

[69] P. P. Kulandaisamy, Z. Abdul-Samad, H. Salleh, A. C. Kasim, H. H. Basri, and M. B. Abdullah, "Critical Success Factors of ICT Projects in Malaysia," 2025. [Online]. Available: https://www.jisem-journal.com/

[70] H. bin M. Yusof, D. bin M. Yusof, and N. binti M. Adnan, "The Role of the Principal-Agent-Client Model in Understanding Corruption in the Public Procurement Sector in Malaysia," Intellectual Discourse, vol. 32, no. 1, pp. 189–212, 2024, doi: 10.31436/id.v32i1.2026.

[71] D. Schlegel, K. Schuler, and J. Westenberger, "Failure factors of AI projects: results from expert interviews," International Journal of Information Systems and Project Management, vol. 11, no. 3, pp. 25–40, 2023, doi: 10.12821/ijispm110302.

[72] A. S. T. Wahyudi, T. Raharjo, and L. J. Wantania, "Risk management for energy efficiency in information technology project: A case of a government agencies in Indonesia," in IOP Conference Series: Earth and Environmental Science, IOP Publishing Ltd, Feb. 2022. doi: 10.1088/1755-1315/969/1/012056.

[73] S. Mohammad and A. Javid, "Digital Transformation And Administration Innovation Decision-Making Framework for Evaluating Outsourcing Strategies in Industrial Projects," 2024.

[74] [74] Kahtan, and U. Kebangsaan Malaysia, "Critical Success Factors for Knowledge Transfer in Enterprise System Projects: A Theoretical and Empirical Investigation Institute of IR 4.0 (IIR4.0)," 2025. [Online]. Available: www.ijacsa.thesai.org

[75] J. Oranga and A. Matere, "Qualitative Research: Essence, Types and Advantages," OAlib, vol. 10, no. 12, pp. 1–9, 2023, doi: 10.4236/oalib.1111001.

[76] M. M. AlHajri, "Exploring Information System Disruptions in Government Operations," IEOM Society International, Jun. 2024. doi: 10.46254/an14.20240154.

Empowering Educators to Develop Augmented Reality Applications by Integrating Computational Thinking and Experiential Learning

Najwa Abd Ghafar[1], Nazatul Aini Abd Majid[2]*, Mohd Nor Akmal Khalid[3],
Shinobu Hasegawa[4], Tan Siok Yee[5], Lam Meng Chun[6]
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia[1, 2, 3, 5, 6]
College of Computing-Informatics and Mathematics, Universiti Teknologi MARA (UiTM), Segamat, Johor, Malaysia[1]
Center for Innovative Distance Education and Research, Japan Advanced Institute of Science and Technology (JAIST), Nomi, Japan[4]

***Abstract*—Augmented reality (AR) promises rich, immersive experiences in learning, yet many educators lack the technical expertise to develop AR content independently. Existing authoring environments often emphasise functionality over instructional design, leaving educators reliant on pre-built resources that seldom match their learning goals. This study proposes EduCATE-AR, a template-based authoring model that integrates Experiential Learning (EL) and Computational Thinking (CT) to embed pedagogies directly into a modular-based authoring approach. Using a three-phase design science research approach, pedagogical principles were mapped onto AR authoring workflows to develop the model. A template-based editing environment was then constructed based on these mappings, and its use was evaluated through a mixed-methods study involving a focus group with educators. Quantitative results based on the Technology Acceptance Model (TAM) indicated strong perceived usefulness and intention to use, particularly among educators without prior AR development experience. Qualitative findings further highlighted the model's value in reducing technical barriers and supporting pedagogical structuring, while also revealing the need for improved extensibility and subject-specific flexibility. Collectively, the proposed model is a valuable addition to AR instructional design by empowering educators as content creators to develop pedagogically aligned AR applications with greater autonomy.**

***Keywords*—Augmented reality; content creation; instructional design; template-based authoring; experiential learning; computational thinking**

I. Introduction

With the advancement of technologies, digital-native learners are expecting visually rich and interactive educational content, pushing educators to adopt more engaging instructional strategies. Among the possible technologies, Augmented Reality (AR) particularly stands out for its ability to overlay real-world environments with dynamic digital content, offering immersion in the learning experiences [1-4] to improve motivation and interests [5-9]. While AR is often accessed through mobile devices and marker-based tracking [10, 11], its adoption tends to prioritise engagement. Learners may initially be engaged; however, feature-driven applications can result in superficial engagement, lacking the depth required for sustained knowledge acquisition [12]. Hence, a critical challenge persists: how can AR experiences be designed to achieve not only engagement but also pedagogically meaningful learning?

Developing AR applications in education is not without its limitations. Studies in AR authoring often emphasise technical capabilities in interactivity but pay less attention to pedagogical structuring, which places usability barriers and limits educators' involvement in shaping meaningful AR learning experiences [13-15]. The disconnection of pedagogical relevance in AR content is a concern echoed in recent reviews of AR adoption in education [16]. Although approaches like inquiry-based and situated learning have been constantly explored within AR [17-19], there is limited integration of structured pedagogies like Experiential Learning (EL) and Computational Thinking (CT) in providing support for AR design that emphasises both reflective thinking and systematic problem-solving.

Integrating these structured pedagogies can provide a foundation for enhancing the educational impact of AR [20] by moving beyond surface-level engagement toward deeper learning outcomes. EL supports reflective learning through a cycle of experience, reflection, conceptualisation, and application [21-23], which aligns naturally with AR's immersive potential. CT, on the other hand, fosters essential cognitive skills through processes such as pattern recognition, decomposition, abstraction, and algorithmic reasoning [24-26], providing a valuable approach for designing logic-driven AR learning tasks. Although EL and CT have been applied separately in AR learning environments [24, 27, 28], their combined potential within a unified AR content creation framework remains largely unexplored.

To address these challenges, this study proposes a template-based authoring model designed for educators with limited technical expertise. The model integrates EL and CT principles directly into the AR content creation workflow, embedding pedagogical structure into modular templates. The study employed a focus group methodology to evaluate the model's pedagogical alignment and usability, gathering feedback through open-ended reflections and Technology Acceptance Model (TAM)-based questions. Specifically, the research investigates: 1) how EL and CT can be systematically mapped into an AR authoring model (RQ1), 2) how prior AR development experience influences educators' perceptions of

*Corresponding author.

the model (RQ2), and 3) which features are most valued, and what refinements are considered necessary for effective adoption (RQ3).

The remaining sections of this study have been structured in the following manner. Section II provides a review of related work on AR applications in education and their pedagogical underpinnings. Section III outlines the research design, introduces the proposed template, and defines participant recruitment, focus group procedures, and data analysis approach. Sections IV and V present the findings and discussion from the focus group discussions, examining educators' reflections on pedagogical needs, perceived challenges, and opportunities for AR content creation. Finally, Section VI provides the conclusion, which shows the study's limitations and directions for future research.

II. Related Works

The potential of AR to enhance learners' engagement and encourage positive learning attitudes has received considerable attention in the field of education. Even though current studies are largely centered on demonstrating AR's advantages for learners, a growing body of literature is starting to shift towards the involvement of educators as primary creators of AR content. Despite this shift, much of the existing literature leans towards the technical capability of authoring tools, or their usability from a technical perspective [13, 29], omitting clear guidance on aligning AR activities with learning objectives.

A. Authoring Environment for AR Content Creation

AR authoring environments vary widely in both their level of technical abstraction and the extent of pedagogical support they offer, as depicted in Table I. Three recurring patterns emerge from the literature. Firstly, based on authoring style, visual editing environments help in lowering the entry barrier for educators [24, 30, 31]. Template-based editing can also accelerate content reuse [32, 33], yet in practice, most of these tools often produce static overlays with limited adaptability. As for pedagogy integration, it is explicitly seen in ExposAR [24], where learners are engaging in CT concepts during AR activities. However, none of the reviewed authoring environments systematically integrates the EL cycle to guide lesson sequencing. Moreover, since engagement with AR content is deemed important [34, 35], many authoring tools still offer restricted interactivity, missing opportunities for real-time manipulation and learner-driven exploration [36-38].

Gaps in AR authoring environments become more apparent when they are seen to reduce programming load on educators, but are mostly still privileging technical usability over a structured, theory-driven approach to AR design. For instance, Paper Trail [37] and WebArt [36] offer immersive experiences with tangible artefacts and web-based capabilities, respectively, but their main focus remains on interaction mechanics and visual augmentation, rather than on embedding a pedagogical framework into the flow of activities. Therefore, this investigation addresses the need to provide educators with a low-effort, high-guidance AR content creation process that integrates EL and CT while preserving the learner-driven interactivity that mobile AR affords. The EduCATE-AR model addresses this shortfall by embedding pedagogical concepts drawn from EL and CT directly into each authoring step.

TABLE I. Specifications of Existing AR Authoring Tools

Ref.		[32]	[39]	[40]	[33]	[36]	[31]	[41]	[30]	[38]	[24]	[37]
Authoring Style	Template Editing	✓	✓	✓	✓	✓						
	Visual Editing						✓	✓	✓	✓	✓	✓
Pedagogy	Situated Learning	✓										
	Learning- while-making									✓		
	Distance Learning		✓					✓				
	Computational Thinking										✓	
Interactivity	Static		✓	✓	✓			✓	✓			
	Dynamic	✓				✓	✓			✓	✓	✓

B. Integration of Experiential Learning and Computational Thinking in AR Content

EL involves a continuous four-stage cycle that facilitates knowledge construction through experience and application [42]. AR can strengthen this cycle by providing immersive environments, real-time feedback, and interactive content [21, 43, 44], hence connecting hands-on activities with cognitive learning [45]. CT, by contrast, is a series of cognitive processes that uses approaches comparable to those applied in computing [25, 26, 46]. When applied within AR, CT enables structured problem-solving through tasks such as programming sequences, as demonstrated in the ARQuest game [47]. Together, EL and CT provide a strong pedagogical foundation for AR content, where EL fosters experiential and reflective learning, while CT instils logic and structured reasoning. Their integration transforms AR into a meaningful instructional medium that supports both cognitive growth and hands-on engagement. As noted by Ramli et al. [48], effective AR pedagogy must address the what, why, and how of learning. Hence, the first two are outlined in Table II, whereas the how is clarified through the EL-CT alignment presented in this study.

Researchers have noted persistent inconsistencies between technological capabilities and instructional objectives, resulting in AR applications that may capture students' attention but fall short in promoting deep learning [12, 16, 49]. While EL and CT have demonstrated value in AR, they are rarely embedded within authoring tools, creating a technological–pedagogical

gap that can limit the educational usefulness of AR applications among educators [12]. Moreover, studies have shown that AR adoption is shaped by factors beyond technical features. The MARAM model by Koutromanos et al. [50] extends TAM to incorporate dimensions such as perceived usefulness, ease of use, facilitating conditions, and self-efficacy, highlighting their influence on AR adoption by educators.

TABLE II. THEORETICAL GROUNDING OF EXPERIENTIAL LEARNING AND COMPUTATIONAL THINKING IN AR

Approach	Explanation		Ref.
Experiential Learning	What	A learning theory that promotes active knowledge construction through immersive and interactive AR activities, allowing learners to engage directly with simulated environments.	[51, 52]
	Why	Enhance motivation and engagement in AR by enabling learners to explore digital representations of real-world phenomena, make personal discoveries, and reflect on their experiences within the AR space.	[28, 42, 43]
Computational Thinking	What	A cognitive approach to solving problems in digital contexts.	[25, 26, 46, 53]
	Why	Fosters critical thinking skills and 21st-century competencies through structured, logic-based learning in interactive environments.	[25, 54]

Hence, by integrating EL and CT into a single template-based design, this study responds directly to calls for pedagogically aligned AR design while preserving ease of use for educators. It underpins the investigation of how the authoring model supports pedagogical mapping, its acceptance across varying levels of background expertise, and how users identify helpful features and areas for improvement, ensuring that the model meets educators' needs and further advances the AR's potential for adoption.

III. METHODOLOGY

The study followed a three-phase design science research sequence that progresses from solution design to artefact construction and empirical evaluation, as illustrated in Fig. 1. Phase 1 encompasses the authoring model that maps EL phases with complementary CT skills to mobile AR interactions, thereby addressing RQ1. Phase 2 translated the model design into the form of a working authoring template, demonstrating the feasibility of implementing the EL-CT alignments for AR content creation. Phase 3 gathered educators' perceptions through a focus-group study, generating evidence for RQ2 and RQ3. Consistent with design science research, in which empirical evaluation may take the form of satisfaction and perception-based feedback [55]. The study's evaluation was scoped to perceived usefulness and usability prior to performing any classroom-based effectiveness assessment. This practice ensures that each research question is examined at the stage most suited to its focus and that later insights remain anchored in the established conceptual groundwork.

A. Alignment of Experiential Learning and Computational Thinking in AR Content

The EduCATE-AR model is grounded in EL, comprising four phases, which are Concrete Experience (CE), Reflective Observation (RO), Abstract Conceptualisation (AC), and Active Experimentation (AE). Each phase is paired with a CT practice and a cognitive task: 1) Pattern Recognition (PR) and Identifying for CE, 2) Abstraction (AB) and Interpreting for RO, 3) Decomposition and Relating for AC, and (4) Algorithmic Thinking (AL) and Sequencing for AE, as presented in Fig. 2. Table III details how each module's AR interaction supports its cognitive task, where the complexity of that cognition is shaped by the learning activity's design. For instance, at CE, scale-and-rotate can range from recognising a single feature to a more complex spatial analysis across various viewpoints. Tap-to-reveal in RO enables the retrieval of a single contextual annotation, extending to drawing connections across several related details. In AC, scale-and-rotate similarly supports viewing an individual component in isolation, though it can equally demand reasoning about how multiple components function together. Drag-and-drop at AE is capable of simple recall through reordering pre-labelled steps, up to inferring the correct sequence through procedural reasoning.

TABLE III. AR CONTENT DESIGN BASED ON THE LEARNING MODULES

Cognitive-based Module	AR Interaction	Learning Objective in AR
Identifying	Two-finger (scale-and-rotate)	*LO1:* Learners observe and explore virtual objects in the AR environment to identify recurring features and build initial familiarity.
Interpreting	One-finger (tap-to-reveal)	*LO2:* Learners extract meaning from their prior observations by accessing contextual annotations on virtual objects that highlight important features.
Relating	Two-finger (scale-and-rotate)	*LO3:* Learners break down AR content into individual components for inspection to understand their functions and relationships within the overall concept.
Sequencing	One-finger (drag-and-drop)	*LO4:* Learners test their ordered actions in a new scenario and observe the outcome.

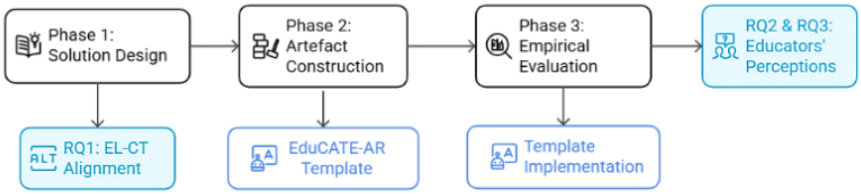


Fig. 1. Research design of the study.

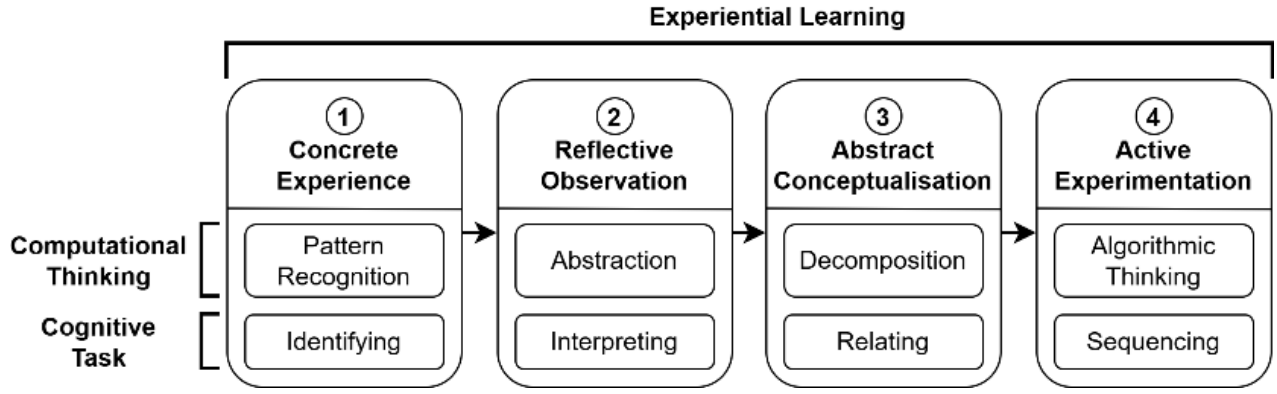


Fig. 2. Experiential learning and computational thinking mapping with cognitive task.

B. Design of the Proposed Authoring Model

As depicted in Fig. 3, the EduCATE-AR model integrates EL phases, CT practices, cognitive tasks, and AR interactions into a template-based authoring flow, addressing RQ1 by demonstrating how these elements can be systematically embedded in content creation. The template layer contains four sequential modules, one per EL phase (CE→RO→AC→AE), each pairing its CT practice, cognitive task, and AR interaction as detailed in Table III. Educators assemble lessons by selecting these modules and configuring their content, while the template handles the technical implementation of each AR interaction.

This template design is realised in Unity. Each of the four modules is packaged as a prefab, which is a ready-made object bundle containing its own scripts, components, and behaviours, as shown in Fig. 4, so it can be placed into a scene and used directly, without being rebuilt each time. Each prefab is linked to a physical marker, a printed image that the AR camera can recognise. When the camera detects a marker, Vuforia identifies it and triggers Unity to load the matching prefab, rendering its AR content at that location in the real-world scene. Once the content appears, Leap Touch captures the learner's gesture, such as rotating, tapping, or dragging, to support interaction with the AR content.

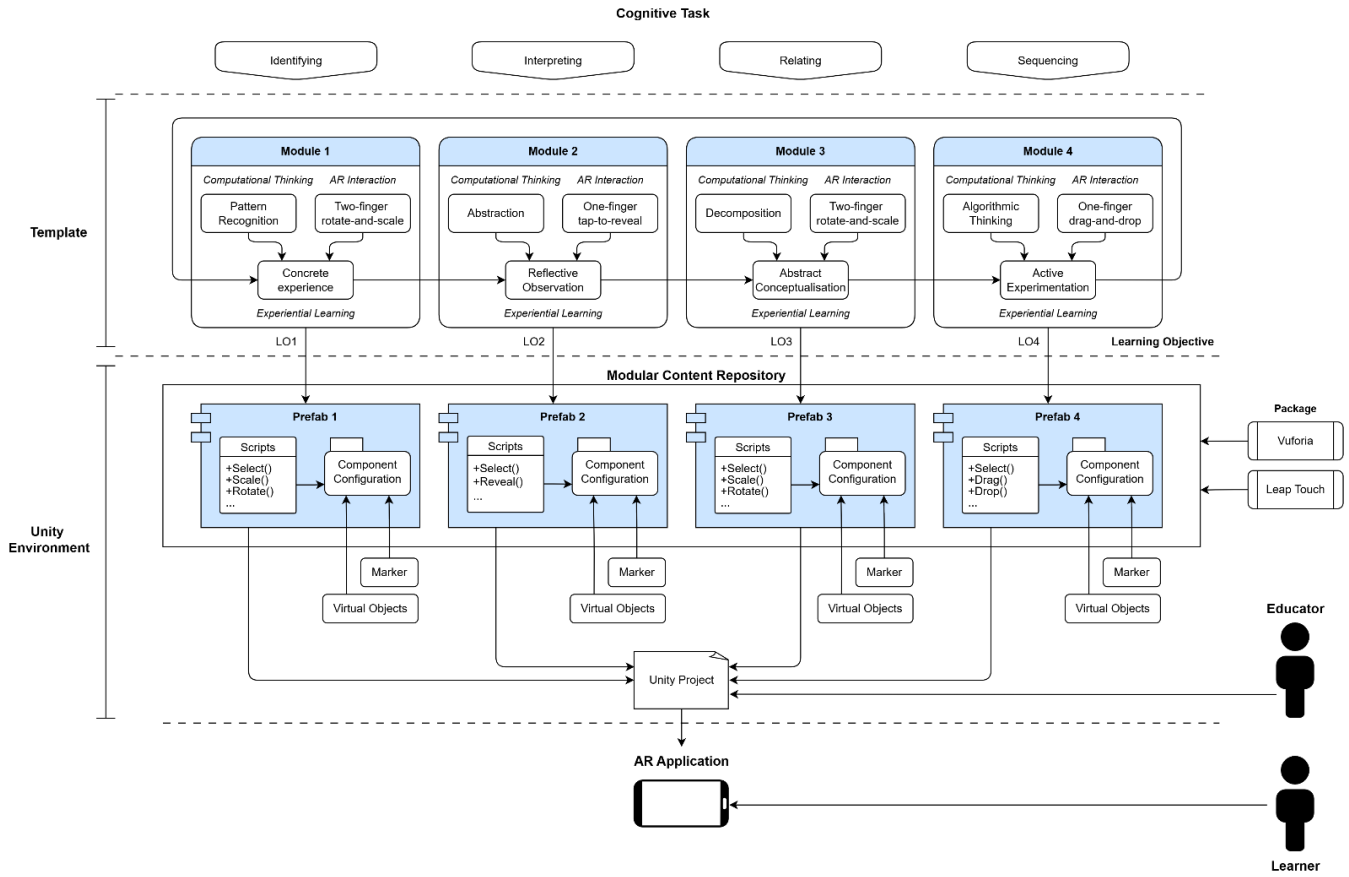


Fig. 3. EduCATE-AR model for template-based AR content creation.

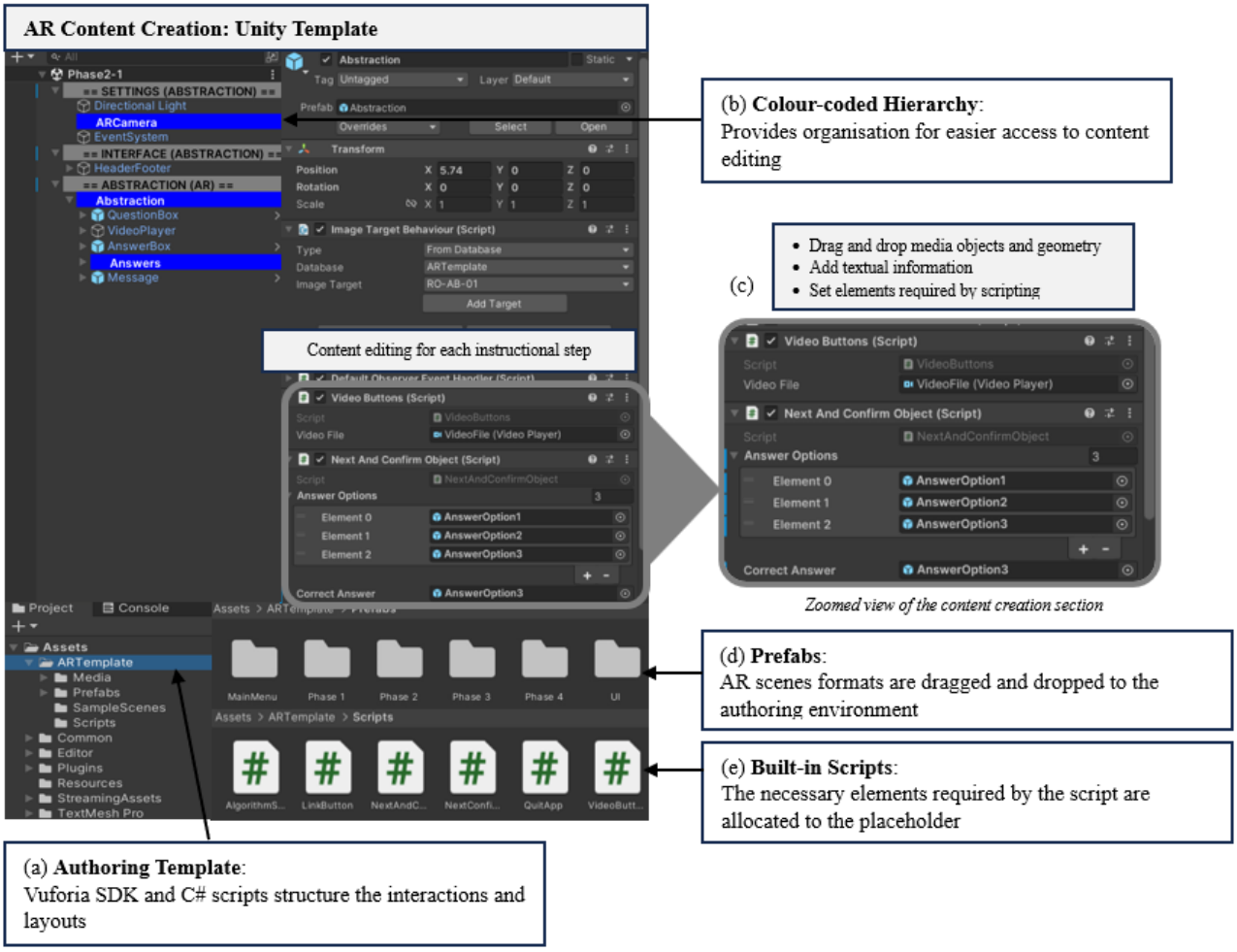


Fig. 4. EduCATE-AR template for educational AR content creation.

Each module in the model is realised as one of four prefabs, matched to its EL phase, CT practice, and cognitive task. Prefab 1 (CE-PR, Identifying) enables open exploration, where learners scale and rotate a virtual object from different angles to spot recurring visual features. Prefab 2 (RO-AB, Interpreting) supports sense-making from those observations, where learners tap to reveal contextual annotations that explain what they are looking at. Prefab 3 (AC-DC, Relating) supports component-level inspection, allowing learners to scale and rotate individual parts of a more complex object to see how each part relates to the whole. Prefab 4 (AE-AL, Sequencing) supports applied problem-solving by providing learners with drag-and-drop elements to be placed in the correct order for testing their understanding in a new scenario. Together, this prefab structure keeps the EL-CT sequence embedded in the authoring process rather than in code, so educators build lessons by selecting and configuring these prefabs instead of programming them directly.

C. Experimental Design and Implementation

This study adopted a focus group evaluation of the EduCATE-AR template. Educators' experiences were examined using TAM [56] and thematic analysis of open-ended reflections, providing a comprehensive view of usability, pedagogical fit, and perceived instructional value [20].

TABLE IV. DISTRIBUTION OF EDUCATORS' DEMOGRAPHIC PROFILE

Demographic (N = 18)	Category	Frequency	Percentage (%)
Gender	Male	5	27.78
	Female	13	72.22
Age	30 - 34	8	44.44
	35 - 39	7	38.89
	40 - 44	1	5.56
	45 - 49	2	11.11
Teaching Subject / Discipline	Computing	11	61.11
	Science	2	11.11
	Language	3	16.67
	Business	2	11.11
AR Development Experience	Experienced	5	27.78
	Non-experienced	13	72.22

1) Participants: Eighteen educators, consisting of school teachers and university lecturers, participated voluntarily and were selected through an open invitation. All participants provided informed consent and were anonymised using unique

identifiers. As shown in Table IV, most participants were female (72.22%) and aged between 30 and 39 (83.33%). The majority taught Computing (61.11%), but most did not have prior AR development experience (72.22%). Based on their profiles, this group was able to offer perspectives from both technical and non-technical backgrounds. A sample size of 18 educators is appropriate for a focus group-based study, as thematic insights are commonly achieved with approximately 12 to 18 participants [57, 58] and prior studies employing similar designs in educational technology contexts have used similar or smaller samples [59, 60].

2) Procedure: A workshop was conducted to introduce participants to key AR concepts, CT and EL pedagogies, and essential Unity workflows. This was followed by hands-on training using the EduCATE-AR template (Fig. 5), where educators created an AR application focused on geometric shapes. The session was facilitated to help them through content development, including sequencing AR modules across EL stages and embedding the corresponding CT practices, so each authoring step targeted a clear learning objective.



Fig. 5. Educators using the EduCATE-AR template during AR content creation.

3) Measurements and variables: A mixed-methods approach was employed to evaluate how well the EduCATE-AR template aligns with educators' pedagogical and technical needs. Quantitative data were collected using an 11-item TAM-based questionnaire measuring Perceived Usefulness (PU), Perceived Ease of Use (PEU), and Intention to Use (ITU) on a 7-point Likert scale. Qualitative data were also gathered through two open-ended questions capturing participants' reflections and suggestions. Thematic analysis was conducted and triangulated with TAM findings to identify converging insights. Additionally, prior AR development experience was examined as a moderating factor influencing perceptions.

IV. Results

This section presents the quantitative and qualitative findings from the focus-group evaluation, covering educators' TAM-based perceptions and the thematic patterns identified in their reflections.

A. Educators' Perceptions Based on Technology Acceptance Model

Descriptive statistics were calculated for each TAM construct to evaluate participants' overall perceptions of the EduCATE-AR template. Internal reliability for all constructs was robust, with Cronbach's alpha values exceeding the recommended threshold: PU ($\alpha = 0.824$), *PEU* ($\alpha = 0.976$), and *ITU* ($\alpha = 0.925$), confirming the reliability of the instrument in capturing educators' perceptions towards the template. As shown in Table V, PU ($M = 6.694$, $SD = 0.425$) emerged as the highest-rated dimension, indicating strong educators' agreement regarding the template's relevance to pedagogical goals and facilitating the development of interactive AR content. This was followed by ITU ($M = 6.333$, $SD = 0.825$), suggesting that educators showed willingness to adopt the template in future practice. PEU received a slightly lower mean score ($M = 6.097$, $SD = 1.473$), reflecting varied individual experiences during template usage. This signals potential challenges in conceptual clarity for some educators, particularly those without prior exposure to Unity or AR development.

TABLE V. Mean (M), Standard Deviation (SD) and Cronbach's Alpha (α) for Items in TAM Constructs

Item	Question	*M*	*SD*	*α*
Perceived Usefulness		**6.694**	**0.425**	**0.824**
PU1	The template helps me create AR applications that meet my teaching goals.	6.611	0.502	
PU2	The template makes it easier to integrate pedagogies in my AR applications.	6.778	0.428	
PU3	The template improves the lesson quality of my AR applications.	6.611	0.608	
PU4	The template lets me create AR applications with interactive elements.	6.778	0.548	
Perceived Ease of Use		**6.097**	**1.473**	**0.976**
PEU1	Learning how to use the template is easy.	6.167	1.505	
PEU2	The template makes it easy to create AR applications with minimal effort.	6.278	1.447	
PEU3	The steps for building an AR lesson with the template are simple.	5.833	1.618	
PEU4	The interface and structure of the template are understandable.	6.111	1.530	
Intention to Use		**6.333**	**0.825**	**0.925**
ITU1	I am interested in using the template for future AR applications.	6.389	0.778	
ITU2	Given the opportunity, I will use the template in my teaching practice.	6.278	0.826	
ITU3	I will use the template to integrate pedagogies in my future AR applications.	6.333	1.029	

To evaluate whether perceptions differed based on educators' prior expertise in AR content creation, a Mann-Whitney U test was conducted for each TAM construct. Prior to group comparison, the Shapiro-Wilk test was used to assess

normality separately within each experience subgroup. As presented in Table VI, all three constructs showed significant deviation from normality within the non-experienced subgroup, while the experienced subgroup did not. Given the non-normality observed in the larger subgroup, non-parametric analysis was adopted for both comparisons.

TABLE VI. SHAPIRO-WILK TEST FOR NORMALITY OF TAM CONSTRUCTS

Variables	Group	Shapiro-Wilk		
		W	*df*	*p*
Perceived usefulness (PU)	Non-experienced	0.513	13	<.001
	Experienced	0.902	5	0.421
Perceived ease of use (PEU)	Non-experienced	0.689	13	<.001
	Experienced	0.894	5	0.377
Intention to use (ITU)	Non-experienced	0.658	13	<.001
	Experienced	0.828	5	0.135

TABLE VII. MANN-WHITNEY U TEST COMPARING TAM SCORES BASED ON EXPERTISE IN AR DEVELOPMENT

Variables	Expertise in AR	*N*	Mean Rank	Total Ranks	*U*	*p*	*r*
Perceived usefulness (PU)	Non-experienced	13	11.46	149.00	7.000	0.010	0.785
	Experienced	5	4.40	22.00			
Perceived ease of use (PEU)	Non-experienced	13	9.19	119.50	28.500	0.703	0.123
	Experienced	5	10.30	51.50			
Intention to use (ITU)	Non-experienced	13	11.04	143.50	12.500	0.046	0.615
	Experienced	5	5.50	27.50			

The application of the Mann-Whitney U test ensured the robustness of inferential comparisons and supported the integrity of the conclusions drawn about educators' differing levels of acceptance. Table VII reports that non-experienced participants give significantly higher scores for PU ($U = 7.000$, $p = 0.010$) and ITU ($U = 12.500$, $p = 0.046$) compared to experienced participants, while PEU did not significantly differ between groups ($U = 28.500$, $p = 0.703$), indicating a shared perception of usability irrespective of prior exposure to AR development. Given the smaller experienced subgroup (n=5), rank-biserial effect sizes were also computed for each comparison. ITU (r=0.615) and PU (r=0.785) showed large effects, suggesting these differences are unlikely to be due to chance from the small subgroup size, while PEU showed a negligible effect (r=0.123), consistent with its non-significant result. These findings indicate that non-experienced participants rated the template-based design more favourably than experienced participants, particularly on PU and ITU, reflecting strong initial acceptance among less technical users within this preliminary evaluation. Collectively, these findings reinforce the potential of the EduCATE-AR template, particularly for novice AR creators seeking simplified authoring pathways.

B. Thematic Insights from Educators' Reflections

Responses were manually coded using predefined categories mapped to TAM's three core constructs, which are PU, PEU, and ITU. Fig. 6 presents the thematic breakdown and frequency based on the qualitative educators' responses. In terms of PU, educators highlighted that the low technical barrier (f = 6) of the EduCATE-AR template made it suitable for non-programmers and novice AR content creators. One educator noted, "The template helps people who've never made an AR app before, especially for teachers who are not good with technical things", proving how the template opens up AR content creation to a broader, less technical audience. This reinforces its accessibility for educators who typically struggle with coding-based platforms. Pedagogical structuring (f = 4) was also appreciated for simplifying AR development. This was captured in remarks like, "All is suitable in the template and helps navigate my application well from one module to another", suggesting that the template's design included built-in sequences to support AR lesson progression. This aligns with the goal of helping educators structure their lesson content systematically. Moreover, some educators expressed a desire for scalability and expandability (f = 3), particularly with multimedia support. One respondent mentioned, "It's good for now, but I'd like to be able to add videos or maybe audio in future applications", signaling the potential of the template to support more immersive experiences if equipped with richer media capabilities. Also, the template was seen to accelerate development (f = 2), with the support of a comment, "Because it is easy to use the template, maybe I can make AR app faster". This suggests that the predefined structure in the template helped users move from concept to implementation quickly.

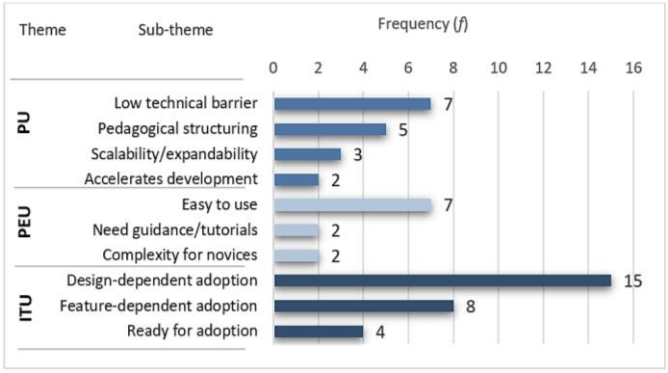


Fig. 6. Frequency of sub-themes across educators' responses.

Regarding PEU, the template's ease of use was the most cited benefit, with easy to use (f = 7) emerging as the dominant sub-theme. A participant remarked, "Everything inside the template is easy to understand and access," highlighting the structure and labelling of components in the template's hierarchy, which eases the process of AR content creation. Nevertheless, not all experiences were equally straightforward. Under need guidance or tutorials (f = 2), one educator suggested improvements, such as a step-by-step walkthrough. This was reflected in the sentiment that "The template is beneficial, but requires different paces among users," pointing to the necessity

of tailored guidance based on their experience levels. Additionally, complexity for novices (f = 2) surfaced through the same concern, suggesting that while the template is understandable for some, others may still find the authoring steps confusing without additional help.

Within the ITU construct, design-dependent adoption (f = 15) emerged as the most prevalent theme, with educators noting that although the template is promising, adoption depends on improved structure, layout, or navigational flow. One participant commented, "Maybe improve the hierarchy and visual layout of the template, so it can be easier to organize the content, especially if I can use it to apply speech or sensor inputs in AR", suggesting how usability limitations in the template's design can constrain adoption. Another group of educators expressed feature-dependent adoption (f = 8), emphasizing that their intent to use the template depends on the inclusion of more engaging, interactive functionalities. As one educator noted, "Add more interactions to make it more engaging", highlighting how richer media elements and interaction capabilities are essential for enhancing learners' attention. These reflections indicate that pedagogical value in AR content is often tied to the need for improved layout and content organisation, as well as expanding the template's interactive capabilities to better engage learners. In contrast, a smaller group of educators demonstrated readiness for adoption (f = 4), with no further requests for template improvement. This is shown in responses like "I think it's good to go for now. After I try to adapt it to my project, I'll see what else is needed", pointing to satisfaction with the template's features and potential willingness to adopt.

V. Discussion

A. Usability in AR Content Creation

Educators' experiences with the EduCATE-AR template underscored the central role of accessibility in AR content creation. Participants consistently highlighted that the template lowered the technical barrier, especially for those without programming experience. The drag-and-drop mechanics and visual organisation of the template, such as colour-coded hierarchies and prefab-based modules, allowed them to focus on content design rather than technical mechanics. This confirms prior findings that *simplified interfaces* enable broader participation in digital content authoring [13, 61]. While the majority of educators described the template as "easy to understand", a few expressed the need for more structured walkthroughs. These comments reflect a broader expectation among educators that authoring platforms should provide *facilitating conditions*, especially for those with minimal prior exposure to AR development. Offering in-app tutorials, visual cues, or sample scenes allows educators to build confidence and avoid trial-and-error inefficiency [50].

Notably, statistical analysis revealed that less experienced participants conveyed significantly higher scores for PU and ITU than their more experienced counterparts. This suggests that template-based editing with low technicalities may be particularly valuable for novice AR creators, providing a structured yet flexible environment that increases PEU. Interestingly, the absence of significant differences in PEU across experience levels further underscores the platform's universal usability. In particular, the ability to use pre-configured prefabs was seen to accelerate development, with educators noting that they could replicate and customise elements quickly, reducing preparation time. Hence, *fast content assembly* was highly valued for accelerating AR application development.

B. Structuring AR Content for Pedagogical Alignment

Beyond usability, participants underlined the importance of *pedagogical structuring* within the AR development process. Educators appreciated how the template supported lesson progression through built-in sequencing based on EL phases across content modules. As noted by one participant, "it helps navigate my application well from one module to another", indicating that this structuring was especially helpful in supporting clear lesson design, as educators were able to align AR content to predefined learning objectives. Prior work on AR environments has highlighted similar findings, where logical sequencing of authoring steps should reflect the instructional logic recognised by educators to reinforce conceptual clarity and instructional control [62].

However, some participants emphasised that pedagogical structuring must be responsive to subject-specific needs. While the EL and CT integration in the template was seen as generally effective, some educators noted that a uniform instructional framework may not always translate well across disciplines. For instance, one language educator noted that "using one pedagogy for all subjects may not be suitable, especially for activities involving pattern recognition or algorithms". A science educator raised a similar point, stating that "for subjects like science that have abstract concepts, using activities that focus only on computational thinking might not be suitable". These responses highlight a broader issue, indicating that although CT can be useful for procedural thinking and problem-solving in technical fields, the AR content design also needs to be adjusted in subjects where conceptual understanding or interpretive thinking is more important [63, 64]. This means that the authoring platform should support diverse interaction modes and pedagogies to ensure that AR content can be tailored to reflect the instructional logic of different disciplines.

Beyond subject-specific responsiveness, *scalability* also depends on how efficiently new content can be authored and reused as adoption grows. Because each module's scripts and component configuration are decoupled from any particular subject's content, similar interaction logic should be able to be reused across different virtual objects and markers without modifying the underlying template. This is consistent with how template-based editing has been shown to accelerate content reuse in AR authoring [33]. Scalability reduces the technical effort needed to extend the authoring template to new subjects. At larger scale, however, this would likely need to be paired with a shared library of ready-made content assets across common subject areas, an approach consistent with reusable learning object repositories that reduce the cost and effort of producing new educational content [32, 65], rather than requiring each educator to individually source and configure their own virtual objects.

Moreover, a few educators emphasised the importance of *content organisation* within the AR scene, reporting that purposeful placement of instructional elements should be

considered. This is derived from responses such as "improve the visual layout of the content". These insights affirm the necessity for content organisation in AR content design. This is linked with established cognitive load theory, which describes the importance of visual coherence in AR learning environments to avoid extraneous cognitive load [66]. While reducing cognitive distractions is valued by some participants, the significance of *multimedia support* was generally acknowledged to facilitate immersion in AR. Several educators mentioned the need for the integration of diverse media types in AR content to support different instructional methods. As an example, a language educator highlighted that audio is an essential element for demonstrating the sound of Arabic vocabulary. These observations are similar to results from prior studies, showing that embedding rich multimedia is key to improving learner satisfaction [66-68].

C. Towards Personalised Content Creation

Based on the responses collected, several participants appreciated the elements of personalisation and flexibility embedded in the template. As noted by one participant, "Everything inside the template is easy to understand and access, which makes it easier to know how to change the components inside the template." Another participant stated, "I can just switch the given element and use copy and paste with all the elements needed". The responses indicate that the template enables educators to personalise lessons by selecting the entry point in the EL cycle, pairing it with the corresponding CT practice, and configuring the AR interaction to embed. For instance, some educators might begin with an AR-based experience in Phase 1 (CE) and shift to a non-AR approach, such as written classroom exercises in Phase 2 (RO). This flexibility enables educators to design their own instructional content based on specific learning outcomes or teaching goals [69].

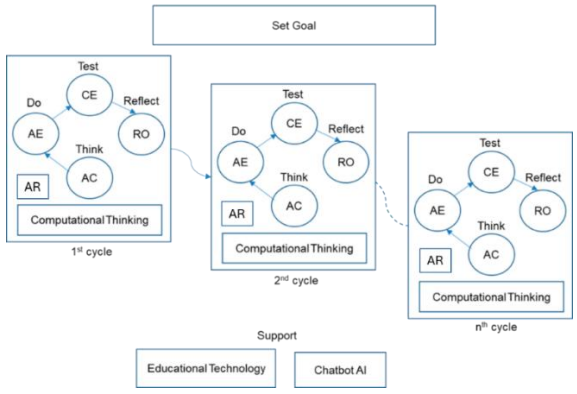


Fig. 7. Additional personalisation layer through Chatbot AI.

Extending this flexibility in content creation, Fig. 7 illustrates how an AI chatbot can add another layer of personalisation to AR content, as a virtual tutor. Since pedagogical agents with human-like assistance have been shown to aid learning [70], such integration can enhance support during the AC phase by providing on-demand explanations or addressing learners' queries with theoretical grounding. Thus, the authoring experience not only allows educators to tailor content flow based on pedagogies but also lays the groundwork for scaffolded and adaptive AR learning experiences, aligned with instructional intent and learner needs [71].

VI. Conclusion

This study investigated how a pedagogically structured integration of CT and EL can empower educators in creating AR learning content. By designing and evaluating the EduCATE-AR model, which operationalises CT principles through the EL cycle, this study demonstrated that educators, particularly those with limited technical expertise, can effectively participate in AR content authoring when supported by a theoretically grounded, template-driven model. The model's foundation in pedagogical mapping between computational logic practice and experiential learning design bridges technical and instructional barriers in AR creation. Quantitative results supported the model's practicality, with novice participants reporting significantly higher levels of PU and ITU. Meanwhile, thematic analysis revealed that the model's modularity and prefab-based design approach were especially valued, though participants also highlighted the need for expanded features and greater pedagogical flexibility to accommodate personalisation.

In terms of limitations, the study was conducted within a focused and relatively small participant group, and all evaluations occurred within a guided research setting rather than a classroom setting. As such, the observed acceptance and positive perceptions may reflect initial exposure rather than sustained integration. Moreover, the study focused primarily on educator perspectives, without directly measuring student learning outcomes or classroom impacts. In addition, the prototype version of the template, while functional, did not offer full customisability or extensibility beyond its prefab modules, which limits insights into more advanced usage scenarios.

Future research should extend these findings by examining how educator-authored AR content affects real classroom environments over time. Longitudinal studies could assess whether the initial motivation and intention to use translate into continued use and improved instructional outcomes. Future evaluations should also incorporate specialised instruments, such as cognitive load assessments, pedagogical alignment rubrics, or CT skill integration metrics, to more directly assess whether the EL-CT theoretical grounding achieves its intended instructional goals, beyond the acceptance-level evidence captured by TAM. Moreover, involving students in future evaluations will be crucial to understanding how educator-authored AR experiences affect engagement, comprehension, and knowledge retention. Expanding the AR authoring template design to support a wider range of pedagogical strategies will also help ensure broader applicability. In addition, performing a controlled comparative study with an existing AR authoring tool would help isolate the specific advantages of embedding EL and CT directly into the authoring workflow. Finally, incorporating adaptive features such as real-time guidance during authoring or AI-based design recommendations could further support educators in aligning their AR content with instructional goals. Such enhancements would not only deepen the educational impact of immersive technologies but also lay the foundation for more personalised, learner-centred AR experiences.

Acknowledgment

The authors acknowledge the Fundamental Research Grant Scheme with grant number FRGS/1/2025/ICT09/UKM/02/1, funded by the Ministry of Higher Education (MOHE), Malaysia.

REFERENCES

[1] A. M. Al-Ansi, M. Jaboob, A. Garad, and A. Al-Ansi, "Analyzing augmented reality (AR) and virtual reality (VR) recent development in education," Social Sciences & Humanities Open, vol. 8, no. 1, p. 100532, 2023.

[2] C. R. Nelson and J. L. Gabbard, "Pedagogical Design Considerations for Mobile Augmented Reality Serious Games (MARSGs): A Literature Review," Electronics, vol. 12, no. 21, p. 4524, 2023.

[3] D. Kamińska et al., "Augmented reality: Current and new trends in education," Electronics, vol. 12, no. 16, p. 3531, 2023.

[4] M. C. Lam, S. M. Lim, and S. Y. Tan, "User Evaluation on a Mobile Augmented Reality Game-based Application as a Learning Tool for Biology," TEM Journal, vol. 12, no. 1, 2023.

[5] M. Ishihara and P. Rattanachinalai, "Learning basic concepts of computer programming with path-finding task in ar and its properties," Education and Information Technologies, pp. 1-24, 2022.

[6] S. C. Steinhaeusser, A. Riedmann, M. Haller, S. Oberdörfer, K. Bucher, and M. E. Latoschik, "Fancy Fruits-An Augmented Reality Application for Special Needs Education," in 2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), 2019: IEEE, pp. 1-4.

[7] Y. Yüzüak and H. Yiğit, "Augmented reality application in engineering education: N-Type MOSFET," International Journal of Electrical Engineering & Education, vol. 60, no. 3, pp. 245-257, 2023.

[8] D. M. E. Asiri, K. H. Allehaibi, and A. H. Basori, "Multimodal Interaction Framework for Collaborative Augmented Reality in Education," IJCSNS International Journal of Computer Science and Network Security, vol. 22, no. 7, 2022, doi: 10.22937/IJCSNS.2022.22.7.33.

[9] B. Baran, Ş. N. Kaptanoğlu, E. Esen, and D. M. Siyez, "Reproductive system augmented reality application for sexual health classes," International Journal of Sexual Health, vol. 32, no. 4, pp. 408-420, 2020.

[10] H. A. Khan, S. Jamil, M. J. Piran, O.-J. Kwon, and J.-W. Lee, "A Comprehensive Survey on the Investigation of Machine-Learning-Powered Augmented Reality Applications in Education," Technologies, vol. 12, no. 5, p. 72, 2024.

[11] D. Roopa, R. Prabha, and G. Senthil, "Revolutionizing education system with interactive augmented reality for quality education," Materials Today: Proceedings, vol. 46, pp. 3860-3863, 2021.

[12] S. Gil Parga, U. Singh, J. Gutierrez, and S. Marks, "Pedagogical design in education using augmented reality: a systematic review," Interactive Learning Environments, vol. 32, no. 8, pp. 4219-4236, 2024.

[13] A. Dengel, M. Z. Iqbal, S. Grafe, and E. Mangina, "A review on augmented reality authoring toolkits for education," Frontiers in Virtual Reality, vol. 3, p. 798032, 2022.

[14] M. M. da Silva, V. Teichrieb, and P. Smith, "What are Teacher's Needs Concerning Augmented Reality Digital Authoring Tools?," in Anais dos Workshops do Congresso Brasileiro de Informática na Educação, 2019, vol. 8, no. 1, p. 1452.

[15] C. Pasalidou and N. Fachantidis, "Teachers' perceptions towards the use of mobile augmented reality: The case of Greek educators," in Internet of Things, Infrastructures and Mobile Applications: Proceedings of the 13th IMCL Conference 13, 2021: Springer, pp. 1039-1050.

[16] T. Mirza, R. Dutta, N. Tuli, and A. Mantri, "Leveraging augmented reality in education involving new pedagogies with emerging societal relevance," Discover Sustainability, vol. 6, no. 1, pp. 1-15, 2025.

[17] H. Jiang, D. Zhu, R. Chugh, D. Turnbull, and W. Jin, "Virtual reality and augmented reality-supported K-12 STEM learning: trends, advantages and challenges," Education and Information Technologies, pp. 1-37, 2025.

[18] A. Kleftodimos, M. Moustaka, and A. Evagelou, "Location-based augmented reality for cultural heritage education: Creating educational, gamified location-based AR applications for the prehistoric lake settlement of Dispilio," Digital, vol. 3, no. 1, pp. 18-45, 2023.

[19] W. Tarng, Y.-J. Lin, and K.-L. Ou, "A virtual experiment for learning the principle of Daniell cell based on augmented reality," Applied Sciences, vol. 11, no. 2, p. 762, 2021.

[20] M. Romano, P. Díaz, and I. Aedo, "Empowering teachers to create augmented reality experiences: the effects on the educational experience," Interactive Learning Environments, vol. 31, no. 3, pp. 1546-1563, 2023.

[21] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," Computers & Education, vol. 147, p. 103778, 2020.

[22] N. Moorhouse, M. C. Tom Dieck, and T. Jung, "An experiential view to children learning in museums with augmented reality," Museum Management and Curatorship, vol. 34, no. 4, pp. 402-418, 2019.

[23] N. M. Noor, M. Ismail, R. L. Yussof, and F. H. Yusoff, "Measuring tajweed augmented reality-based gamification learning model (TARGaLM) implementation for children in tajweed learning," Pertanika J. Sci. Technol, vol. 27, pp. 1821-1840, 2019.

[24] M. S. Lunding, J. E. S. Grønbæk, K.-E. K. Bilstrup, M.-L. S. K. Sørensen, and M. G. Petersen, "ExposAR: Bringing augmented reality to the computational thinking agenda through a collaborative authoring tool," in Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1-14.

[25] A. C. Calderon, D. Skillicorn, A. Watt, and N. Perham, "A double dissociative study into the effectiveness of computational thinking," Education and Information Technologies, vol. 25, no. 2, pp. 1181-1192, 2020.

[26] J. M. Wing, "Computational thinking," Communications of the ACM, vol. 49, no. 3, pp. 33-35, 2006.

[27] T.-C. Huang, C.-C. Chen, and Y.-W. Chou, "Animating eco-education: To see, feel, and discover in an augmented reality-based experiential learning environment," Computers & Education, vol. 96, pp. 72-82, 2016.

[28] N. C. Hashim, N. A. Abd Majid, H. Arshad, H. Hashim, and Z. A. A. Alyasseri, "Mobile augmented reality based on multimodal inputs for experiential learning," IEEE Access, vol. 10, pp. 78953-78969, 2022.

[29] M. Ez-Zaouia, I. Marfisi-Schottman, M. Oueslati, C. Mercier, A. Karoui, and S. George, "A design space of educational authoring tools for augmented reality," in International Conference on Games and Learning Alliance, 2022: Springer, pp. 258-268.

[30] Y. Apaza-Yllachura, A. Paz-Valderrama, and C. Corrales-Delgado, "SimpleAR: Augmented Reality high-level content design framework using visual programming," in 2019 38th International Conference of the Chilean Computer Science Society (SCCC), 2019: IEEE, pp. 1-7.

[31] J. M. Mota, I. Ruiz-Rube, J. M. Dodero, and I. Arnedillo-Sánchez, "Augmented reality mobile app development for all," Computers & Electrical Engineering, vol. 65, pp. 250-260, 2018.

[32] R. Wojciechowski and W. Cellary, "Evaluation of learners' attitude toward learning in ARIES augmented reality environments," Computers & Education, vol. 68, pp. 570-585, 2013.

[33] K. van Lopik, M. Sinclair, R. Sharpe, P. Conway, and A. West, "Developing augmented reality capabilities for industry 4.0 small enterprises: Lessons learnt from a content authoring case study," Computers in Industry, vol. 117, p. 103208, 2020.

[34] S. M. Nizam, R. Z. Abidin, N. C. Hashim, M. C. Lam, H. Arshad, and N. Majid, "A review of multimodal interaction technique in augmented reality environment," Int. J. Adv. Sci. Eng. Inf. Technol, vol. 8, no. 4-2, p. 1460, 2018.

[35] M. C. Lam et al., "Interactive Augmented Reality with Natural Action for Chemistry Experiment Learning," Tem Journal, vol. 9, no. 1, 2020.

[36] E. Liu, S. Cai, Z. Liu, and C. Liu, "WebART: Web-based augmented reality learning resources authoring tool and its user experience study among teachers," IEEE Transactions on Learning Technologies, vol. 16, no. 1, pp. 53-65, 2022.

[37] S. Rajaram and M. Nebeling, "Paper trail: an immersive authoring system for augmented reality instructional experiences," in Proceedings of the 2022 CHI conference on human factors in computing systems, 2022, pp. 1-16.

[38] A. Villanueva, Z. Zhu, Z. Liu, K. Peppler, T. Redick, and K. Ramani, "Meta-AR-app: an authoring platform for collaborative augmented reality in STEM classrooms," in Proceedings of the 2020 CHI conference on human factors in computing systems, 2020, pp. 1-14.

[39] J. Cubillo, S. Martin, M. Castro, and I. Boticki, "Preparing augmented reality learning content should be easy: UNED ARLE—an authoring tool for augmented reality learning environments," Computer Applications in Engineering Education, vol. 23, no. 5, pp. 778-789, 2015.

[40] S. Shekhar, P. Maheshwari, A. Singhal, K. K. Singh, and K. Krishna, "ARComposer: Authoring Augmented Reality Experiences through Text," in Adjunct Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, 2019, pp. 63-65.

[41] C. Lytridis, A. Tsinakos, and I. Kazanidis, "ARTutor—an augmented reality platform for interactive distance learning," Education Sciences, vol. 8, no. 1, p. 6, 2018.

[42] D. A. Kolb, Experiential learning: Experience as the source of learning and development. FT press, 2014.

[43] H.-M. Huang, T.-C. Huang, and C.-Y. Cheng, "Reality matters? exploring a tangible user interface for augmented-reality-based fire education," Universal Access in the Information Society, pp. 1-13, 2021.

[44] N. Mahendru and M. Neo, "Enhancing Student Learning of Multimedia Design Concepts Using Augmented Reality," in 2nd International Conference on Creative Multimedia 2022 (ICCM 2022), 2022: Atlantis Press, pp. 178-191.

[45] N. A. A. Nordin, N. A. Abd Majid, and N. F. A. Zainal, "Mobile augmented reality using 3D ruler in a robotic educational module to promote STEM learning," Bulletin of Electrical Engineering and Informatics, vol. 9, no. 6, pp. 2499-2506, 2020.

[46] N. L. Zaharin, S. Sharif, and M. Mariappan, "Computational thinking: A strategy for developing problem solving skills and Higher Order Thinking Skills (HOTs)," Int. J. Acad. Res. Bus. Soc. Sci, vol. 8, no. 10, 2018.

[47] A. Gardeli and S. Vosinakis, "ARQuest: A tangible augmented reality approach to developing computational thinking skills," in 2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), 2019: IEEE, pp. 1-8.

[48] R. Z. Ramli et al., "Designing a mobile learning application model by integrating augmented reality and game elements to improve student learning experience," Education and Information Technologies, pp. 1-28, 2023.

[49] A. M. C. Lazaro, "An Investigation On The Use Of A Mobile Augmented Reality Application In Chemistry," Future Ed Special Edition: Multidisciplinary Research Journal, vol. 5, no. 1, pp. 37-45, 2018.

[50] G. Koutromanos, A. T. Mikropoulos, D. Mavridis, and C. Christogiannis, "The mobile augmented reality acceptance model for teachers and future teachers," Education and Information Technologies, vol. 29, no. 7, pp. 7855-7893, 2024.

[51] A. Lockey, P. Conaghan, A. Bland, and F. Astin, "Educational theory and its application to advanced life support courses: a narrative review," Resuscitation Plus, vol. 5, p. 100053, 2021.

[52] [52] T. Waseem and F. Aslam, "Educational learning theories & their implications in modern instructional designs," Health Professions Educator Journal, vol. 3, no. 2, pp. 25-31, 2020.

[53] C.-Y. Tsai and Y.-C. Lai, "Design and validation of an augmented reality teaching system for primary logic programming education," Sensors, vol. 22, no. 1, p. 389, 2022.

[54] J. Acevedo-Borrega, J. Valverde-Berrocoso, and M. d. C. Garrido-Arroyo, "Computational thinking and educational technology: A scoping review of the literature," Education Sciences, vol. 12, no. 1, p. 39, 2022.

[55] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," Journal of management information systems, vol. 24, no. 3, pp. 45-77, 2007.

[56] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS Quarterly, pp. 319-340, 1989.

[57] G. Guest, E. Namey, and K. McKenna, "How many focus groups are enough? Building an evidence base for nonprobability sample sizes," Field methods, vol. 29, no. 1, pp. 3-22, 2017.

[58] M. M. Hennink, B. N. Kaiser, and M. B. Weber, "What influences saturation? Estimating sample sizes in focus group research," Qualitative health research, vol. 29, no. 10, pp. 1483-1496, 2019.

[59] G. Peikos and A. Sofianidis, "What is the future of augmented reality in science teaching and learning? an exploratory study on primary and pre-school teacher students' views," Education Sciences, vol. 14, no. 5, p. 480, 2024.

[60] M. Abundez Toledo, G. Ghanem, S. Fine, D. Weisman, Y. M. Huang, and A. A. Rouhani, "Exploring the promise of virtual reality in enhancing anatomy education: a focus group study with medical students," Frontiers in Virtual Reality, vol. 5, p. 1369794, 2024.

[61] M. Ez-Zaouia, I. Marfisi-Schottman, and C. Mercier, "Authoring tools: the road to democratizing augmented reality for education," in Proceedings of the 15th International Conference on Computer Supported Education, 2023, vol. 1, pp. 115-127.

[62] W. Fearne and J. D. Hook, "A service design thinking approach: What are the barriers and opportunities of using augmented reality for primary science education," Journal of Technology and Science Education, pp. 329-251, 2023.

[63] Y. Li et al., "Computational thinking is more about thinking than computing," vol. 3, ed: Springer, 2020, pp. 1-18.

[64] C. Angeli and M. Giannakos, "Computational thinking education: Issues and challenges," vol. 105, ed: Elsevier, 2020, p. 106185.

[65] S. Barnes, F. Li, S. Polyakov, H. Xu, and W. Moen, "A repository for learning objects: Supporting the reuse and repurposing of redesigned courses and their content," Proceedings of the American Society for Information Science and Technology, vol. 45, no. 1, pp. 1-13, 2008.

[66] X. Geng and M. Yamada, "An augmented reality learning system for Japanese compound verbs: study of learning performance and cognitive load," Smart Learning Environments, vol. 7, no. 1, p. 27, 2020.

[67] W. Du, R.-y. Liang, and D. Liu, "Factors influencing school teachers' continuous usage intention of using VR technology for classroom teaching," Sage Open, vol. 12, no. 3, p. 21582440221114325, 2022.

[68] M. Jdaitawi, F. Muhaidat, A. Alsharoa, A. Alshlowi, M. Torki, and M. Abdelmoneim, "The Effectiveness of Augmented Reality in Improving Students Motivation: An Experimental Study," Athens Journal of Education, vol. 10, no. 2, pp. 365-379, 2023.

[69] N. F. A. Zainal, R. Din, N. A. Abd Majid, M. F. Nasrudin, and A. H. Abd Rahman, "Primary and secondary school students' perspective on Kolb-based STEM module and robotic prototype," International Journal on Advanced Science, Engineering and Information Technology, vol. 8, no. 4-2, pp. 1394-1401, 2018.

[70] N. A. H. Basri, R. W. O. Rahmat, H. Madzin, and R. Hod, "A User Study on Virtual Tutor Mobile Augmented Reality from Undergraduate Medical Student Perspectives," Asia-Pacific Journal of Information Technology & Multimedia, vol. 12, no. 1, 2023.

[71] L. Rau, D. C. Döring, R. Horst, and R. Dörner, "Pattern-based augmented reality authoring using different degrees of immersion: A learning nugget approach," Frontiers in Virtual Reality, vol. 3, p. 841066, 2022.

A Bidirectional Cross-Modal Attention Framework for Multimodal Sentiment Analysis

B S Harish, C K Roopa, M S Kendagannaswamy, T S Kaveri*
Department of Information Science and Engineering, JSS Science and Technology University, Mysuru 570006, Karnataka, India

***Abstract*—Multimodal sentiment analysis on social media data presents unique challenges due to label noise, modality conflicts, and class imbalance inherent in annotated image-text datasets. The proposed work presents CLIP-CrossFusion Net, a novel multimodal sentiment analysis framework that integrates the Contrastive Language-Image Pre-Training (CLIP) ViT-B/32 image encoder with a RoBERTa-Base text encoder through a bidirectional cross-modal attention mechanism. The proposed approach addresses three key challenges: 1) rich cross-modal feature interaction via bidirectional attention, 2) noisy label robustness through focal loss with label smoothing, and 3) class imbalance mitigation via targeted class weighting and Mixup augmentation. The study further proposes an extended variant with a colour feature branch, supervised contrastive loss, and deeper cross-modal attention layers. Using the proposed model, comprehensive experiments are conducted on both the MVSA-Multiple (19,600 image-text tweet pairs) and MVSA-Single (4,511 pairs) datasets. On MVSA-Multiple, the proposed model achieves 74.39% accuracy and a weighted F1 of 0.7497. On MVSA-Single, the proposed ensemble of three complementary model variants (base model, colour-contrastive variant, and RoBERTa-Large variant) achieved 75.66% accuracy and an F1 of 0.7517, outperforming strong baselines. These results demonstrate the effectiveness of cross-modal attention fusion combined with noise-robust training strategies for multimodal sentiment analysis.**

***Keywords*—Multimodal sentiment analysis; CLIP; RoBERTa; cross-modal attention; social media; deep learning; transformer**

I. Introduction

Twitter, Instagram, and Weibo generate colossal quantities of multimodal content, posts in which opinions are conveyed in the form of text, images, and video simultaneously. Sentiment analysis of such content can be helpful in brand monitoring, tracking health and politics, as well as financial projections [1]. Text-only strategies are very good; however, they still lack what images are saying and the interaction between text and images.

Combining text and image processing (MSA) is more effective than either of the two, yet there are real challenges. Social media image-text pairs often contain conflicting sentiment, e.g., a tweet may contain a complaint, but the image will appear cheerful. To further complicate matters, the MVSA-Multiple dataset [2] has a lot of label noise, because of disagreements between its three annotators. The most difficult of all cases is the Neutral category, as the neutral sentiment is context-specific and can be misinterpreted as positive or negative.

As an extension of this work, this study introduces CLIP-CrossFusion Net, a new multimodal sentiment analysis model, which has three main contributions:

- A cross-modal attention process in both directions, with text and image features cross-attending in parallel so that each modality can influence the other to be reshaped and then fused.
- A noise-robust training method based on Focal Loss and class-balanced weights, label smoothing, and Mixup augmentation to deal with label noise in MVSA-Multiple and improve the performance of Neutral classes.
- Ablation experiments on variants of models in simple concatenation baselines and full cross-modal transformer architectures, and the way each component contributes to the final output.

The proposed MVSA-Multiple experiments achieve an accuracy of 74.39% and a weighted F1 of 0.7497, which is higher than strong baselines. The Neutral class F1, which is enhanced by the weighting of classes, improves from 0.59 to 0.63.

II. Related Work

A. Unimodal Sentiment Analysis

Sentiment analysis has a rich history dating back to lexicon-based approaches [3]. CNN-based [4] and LSTM-based [5] architectures improved performance, followed by attention mechanisms [6] and transformer-based models such as BERT [7] and RoBERTa [8]. However, unimodal approaches are fundamentally limited for social media content where images carry substantial sentiment information.

B. Multimodal Sentiment Analysis

Early multimodal approaches were based on the concatenation of simple features of CNN images and word embeddings [9]. Graph neural network models like MGNNS represented cross-modal relationships as graphs [10]. The D&R Net [11] was able to determine the cross-modal contrast as well as semantic correlation by decomposition and relational networks. CLMLF [12] used transformer-fusion CNN-supervised contrastive learning. MVSA-Single with cross-modal interaction fusion on SECIF [13] produced good results. ICCI [14] scored 79.33% on MVSA-Single by combining HSV colour cues with semantic features.

*Corresponding author.

Using VGG16, Mask-RCNN, BERT, WCNNE, and Dempster-Shafer theory, Biswas et al. [15] obtained 93.48% on MVSA-Multiple.

C. Vision-Language Pretraining

CLIP [16] proposed contrastive pretraining using 400M image-text pairs, and it trains a shared embedding space with impressive transfer properties. ViT [17] showed image recognition with pure transformer architectures, and DeiT [18] optimized vision transformers with distillation. Such models offer much more detailed representations than the conventional CNN encoders, which is why they are adopted into the proposed framework.

D. Cross-Modal Attention and Colour Features

Deep cross-modal attention was applied to two-stage feature fusion in DTN [19]. The SRC-Model [13] used bidirectional image-text interaction gating and MHA. MIGSIE [13] applied GNN-based text-guided multi-channel modules. ICCI showed that the colour cues presented by images are good sentiment indicators, which drives the colour feature branch. The main task of this work is to combine CLIP joint pretraining with explicit cross-modal attention and colour features.

III. Datasets

A. MVSA-Multiple Dataset

The MVSA-Multiple dataset [2] comprises 19,600 image-text pairs on Twitter that have been annotated by three human raters. The final joint label is obtained by majority voting, and the conflicting pairs are eliminated. It has a large class imbalance: Positive (42.0%), Negative (31.4%), Neutral (26.7%), and inherent label noise due to annotation disagreement. Fig. 1 shows the distribution of the classes in the train, validation, and test splits, with the positive class being predominant and the relatively low representation of the neutral category, which compels the class-weighting approach.

B. MVSA-Single Dataset

The MVSA-Single dataset is a set of 4,511 valid image-text pairs on Twitter (after filtering) annotated by one rater. The extreme imbalance of classes, Positive (59.5%), Negative (30.1%), Neutral (10.4%), is problematic with the neutral category. Hence, a typical 8:1:1 train/validation/test split with stratified sampling was applied. This extreme imbalance is depicted visually in Fig. 2, where positive samples are nearly 60% of the data and neutral samples are only 10.4%. Thus, correct Neutral classification is the most difficult on this dataset. Table I summarises the statistics of both datasets.

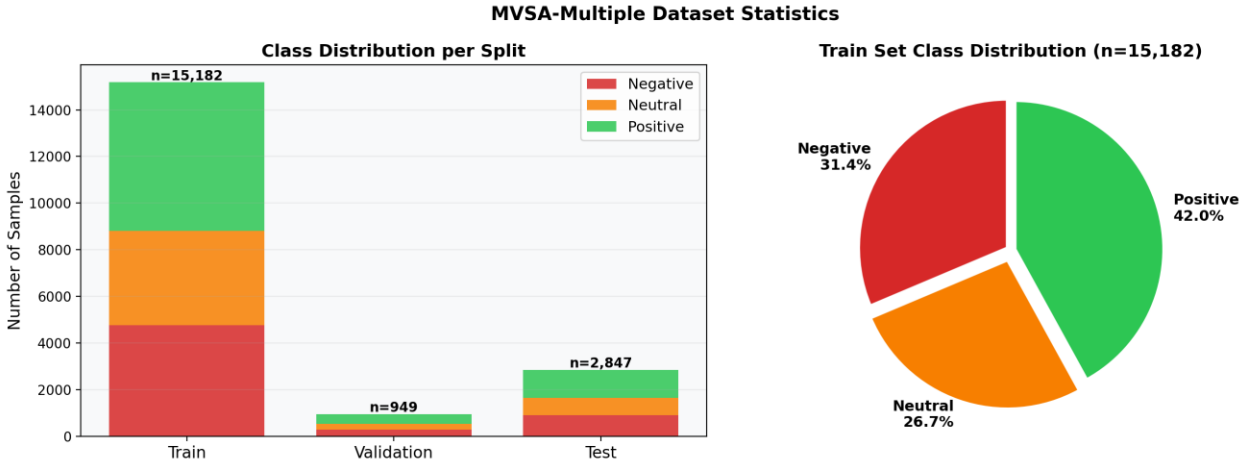


Fig. 1. MVSA-Multiple dataset class distribution across train, validation, and test splits.

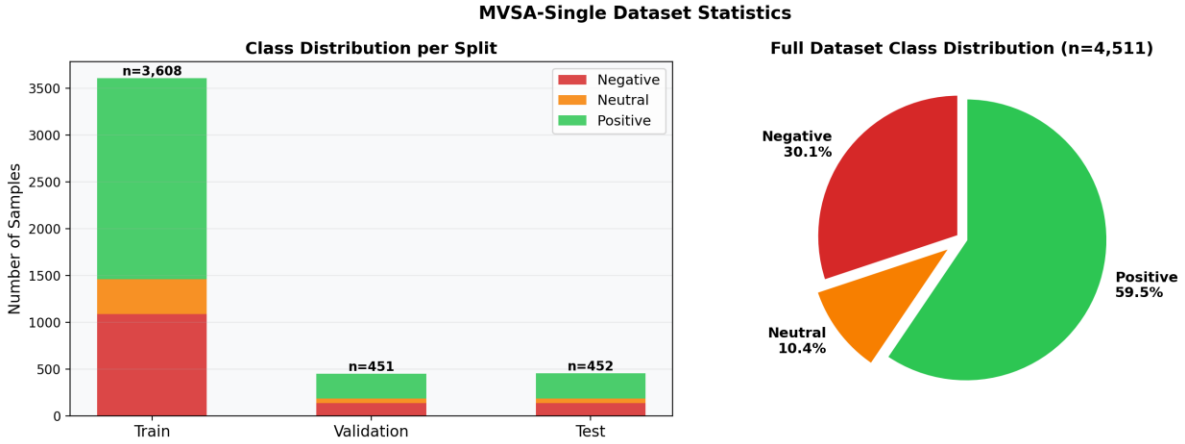


Fig. 2. MVSA-Single dataset class distribution. Severe positive dominance (59.5%) makes neutral classification challenging.

TABLE I. DATASET STATISTICS FOR MVSA-MULTIPLE AND MVSA-SINGLE (ANNOT. = NUMBER OF ANNOTATORS PER SAMPLE)

Dataset	Split	Neg	Neu	Pos	Total	Annot.	Split Ratio
MVSA-Multiple	Train	4,760	4,047	6,375	**15,182**	3	Pre-split
	Val	290	244	415	**949**		
	Test	893	759	1,195	**2,847**		
MVSA-Single	Train	1,086	376	2,146	**3,608**	1	8:1:1
	Val	136	47	268	**451**		
	Test	136	47	269	**452**		

IV. METHODOLOGY

CLIP-CrossFusion Net follows a dual-stream encoder architecture where text and image modalities are independently encoded, projected to a shared embedding space, refined through bidirectional cross-modal attention, and fused via a gated mechanism before classification. The extended colour-contrastive variant adds a colour feature branch that extracts HSV colour histograms from images and integrates them through additional cross-modal attention layers. Fig. 3 presents a complete overview of the architecture, depicting the dual-stream encoders, shared projection layers, bidirectional cross-modal attention module, gated fusion mechanism, and final three-class classifier.

A. Encoders

1) Image Encoder: CLIP ViT-B/32: The proposed approach employs the Vision Transformer ViT-B/32 component of CLIP [16], pretrained on 400 million internet-collected image-text pairs. The encoder produces a 512-dimensional embedding. All layers are frozen except the final two to four transformer blocks, which are fine-tuned with a small learning rate to adapt visual representations to sentiment-relevant features, as expressed in (1).

$$f_{img} = \text{CLIP}_{\text{ViT}}(x_{img}) \in \mathbb{R}^{512} \quad (1)$$

2) Text Encoder: RoBERTa-Base / RoBERTa-Large: For text encoding, RoBERTa-Base [8] is employed for the base, cross-modal attention, focal-loss, and colour-contrastive variants, or RoBERTa-Large for the large-encoder variant. Tweet text is tokenised to a maximum of 128 tokens. The [CLS] token is extracted as the sentence-level representation, as given in (2). The same selective fine-tuning strategy applies: all layers are frozen except the last two to four blocks.

$$f_{text} = \text{RoBERTa}([\text{CLS}], x_{text}) \in \mathbb{R}^{768} \text{ or } \mathbb{R}^{1024} \quad (2)$$

B. Modality Projection

Both modalities are projected to a common 256-dimensional space using separate projection heads with LayerNorm and GELU activation, as defined in (3) and (4).

$$p_{img} = \text{LayerNorm}(\text{GELU}(W_{img} \cdot f_{img})) \in \mathbb{R}^{256} \quad (3)$$

$$p_{text} = \text{LayerNorm}(\text{GELU}(W_{text} \cdot f_{text})) \in \mathbb{R}^{256} \quad (4)$$

C. Bidirectional Cross-Modal Attention

The core innovation is the bidirectional cross-modal attention module enabling each modality to attend to the other simultaneously, as formulated in (5) and (6) for the image-to-text and text-to-image directions, respectively. Two to three such layers are stacked for hierarchical cross-modal reasoning.

$$a_{img} = \text{LayerNorm}(p_{img} + \text{Dropout}(\text{MHA}(p_{img}, p_{text}, p_{text}))) \quad (5)$$

$$a_{text} = \text{LayerNorm}(p_{text} + \text{Dropout}(\text{MHA}(p_{text}, p_{img}, p_{img}))) \quad (6)$$

D. Colour Feature Branch

Inspired by ICCI [14], which demonstrated that colour information improves sentiment analysis, the extended variant introduces a dedicated colour branch. HSV colour histograms (32 bins each for the H, S, and V channels, giving a 96-dimensional feature) are extracted from each image and processed through a two-layer MLP, as given in (7) and (8). Bidirectional cross-modal attention between colour and image features, defined in (9) and (10), allows the model to leverage emotional colour cues.

$$c = \text{concat}(\text{hist}_H, \text{hist}_S, \text{hist}_V) / \Sigma \in \mathbb{R}^{96} \quad (7)$$

$$p_{colour} = \text{ColourBranch}(c) \in \mathbb{R}^{256} \quad (8)$$

$$p_{img} = \text{Attn}_{col \to img}(p_{img}, p_{colour}) \quad (9)$$

$$p_{colour} = \text{Attn}_{img \to col}(p_{colour}, p_{img}) \quad (10)$$

E. Gated Fusion

The base, cross-modal attention, and focal-loss variants use dual-stream gated fusion, while the colour-contrastive and large-encoder variants use triple-stream fusion including colour features, as expressed in (11)-(13). The fused representation is 512-dimensional for the dual-stream variants and 768-dimensional for the colour-contrastive variants.

$$\text{fused} = \text{concat}(a_{img}, a_{text} [, p_{colour}]) \quad (11)$$

$$\text{gate} = \text{Sigmoid}(W_{gate} \cdot \text{fused}) \quad (12)$$

$$z = \text{fused} \odot \text{gate} \quad (13)$$

F. Training Strategy

1) Focal loss with label smoothing: Standard cross-entropy fails on MVSA due to label noise and class imbalance. Focal Loss with label smoothing and class-balanced weights targeting the hard Neutral class is used, as given in (14).

$$L_{FL} = -\Sigma_i \, w_i \cdot (1 - p_i)^{\gamma} \cdot y_i^{smooth} \cdot \log(p_i) \quad (14)$$

Here $\gamma = 2.0$, label smoothing $\varepsilon = 0.1$, and class weights are [Neg: 1.2-1.5, Neu: 1.8-2.0, Pos: 1.0]. The Neutral class receives higher weight to improve recall. The focal term $(1 - p_i)^{\gamma}$ down-weights easy samples.

2) Supervised contrastive loss: The colour-contrastive variant adds a supervised contrastive loss term that pulls same-class embeddings closer while pushing different-class embeddings apart, improving class separation especially for the ambiguous Neutral class. The contrastive term and the combined objective are given in (15) and (16).

$$L_{contrast} = -\text{mean}_{anchors}\ [\ \Sigma(\text{positive pairs}) / \Sigma(\text{all pairs})\ ] \quad (15)$$

$$L_{total} = L_{focal} + \lambda \cdot L_{contrast}, \quad \lambda = 0.05 \quad (16)$$

3) Mixup augmentation: Mixup [20] is applied to the image modality with $\lambda \sim \text{Beta}(0.2, 0.2)$, creating mixed image-label pairs that prevent overly certain predictions on noisy labels.

4) Optimisation: AdamW [21] is used with layer-wise learning rates: encoders (3-5e-6) and fusion head (8e-5 to 1e-4). Training uses cosine annealing with 10% warmup, gradient clipping (max = 1.0), and gradient accumulation (steps = 2-4, effective batch = 64). Early stopping patience is 5-6 epochs on validation F1. The complete hyperparameter configuration of each model variant is listed in Table II.

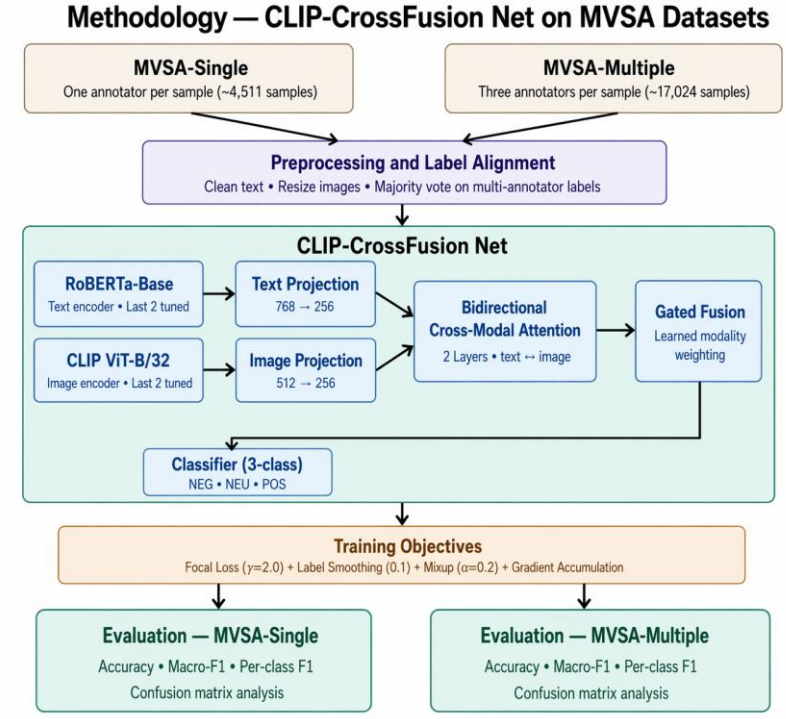


Fig. 3. CLIP-CrossFusion Net architecture: dual-stream encoders, projection, bidirectional cross-modal attention, gated fusion, and 3-class classifier.

TABLE II. HYPERPARAMETER CONFIGURATION PER MODEL VARIANT

Hyperparameter	Base / X-Attn / Focal	Colour + Contrastive	RoBERTa-Large Variant
Image encoder	CLIP ViT-B/32	CLIP ViT-B/32	CLIP ViT-B/32
Text encoder	RoBERTa-Base	RoBERTa-Base	RoBERTa-Large
Colour branch	No	Yes (96-dim)	Yes (96-dim)
Contrastive loss	No	Yes (w = 0.05)	Yes (w = 0.05)
Attention layers	2	3	3
Unfrozen layers	2	4	4
Focal loss γ	2.0	2.0	2.0
Class weights (Neg/Neu/Pos)	1.2 / 1.8 / 1.0	1.3 / 2.0 / 1.0	1.3 / 2.0 / 1.0
Hardware	T4 15 GB	T4 15 GB	T4 15 GB

G. Multi-Model Ensemble

For MVSA-Single, three complementary model variants are combined through weighted probability averaging, as given in (17). The base model provides strong text representations, the colour-contrastive variant adds colour information, and the RoBERTa-Large variant contributes larger text features. The optimal weights (base: 0.5, colour-contrastive: 0.4, RoBERTa-Large: 0.1) are determined by grid search on the validation set. Test-Time Augmentation (three transforms: original, horizontal flip, centre crop) is applied for each model before ensembling.

$$P_{ens} = 0.5 \cdot P_{base} + 0.4 \cdot P_{colour} + 0.1 \cdot P_{large} \quad (17)$$

V. EXPERIMENTS

A. Experimental Setup

All experiments use PyTorch with mixed-precision training (FP16) on an NVIDIA Tesla T4 (15 GB VRAM). For the MVSA-Multiple dataset, the predefined train/validation/test splits are used. For MVSA-Single, 8:1:1 stratified random splits are used. Accuracy and weighted F1-score are reported as primary metrics, consistent with prior work.

B. Baselines

- DeiT-Base + BERT-Base (Concat): primary strong baseline using concatenation fusion.
- ViT-Base + BERT-Base (Concat): Vision Transformer with BERT.
- Swin-Base + BERT-Base (Concat): Swin Transformer with BERT.
- RoBERTa-Large + CLIP ViT-L/14 (large-encoder baseline): larger encoders with gated fusion.
- CLIP ViT-B/32 + RoBERTa-Base (cross-modal attention / focal-loss variants): cross-modal attention variants.
- ICCI (2023) [14]: cross-modal attention with colour cues; 79.33% on MVSA-Single and 73.29% on MVSA-Multiple.
- CLIP-CrossFusion with colour + contrastive (ours): with colour branch and contrastive loss.
- Ensemble of base + colour-contrastive + RoBERTa-Large (ours): best ensemble on MVSA-Single.

VI. RESULTS AND DISCUSSION

A. MVSA-Multiple Results

Table III presents results on MVSA-Multiple. The proposed CLIP-CrossFusion Net with Focal Loss and class weighting achieves 74.39% accuracy and 0.7497 weighted F1, outperforming all baselines including ICCI (73.29%). Fig. 4 visualises the comparison.

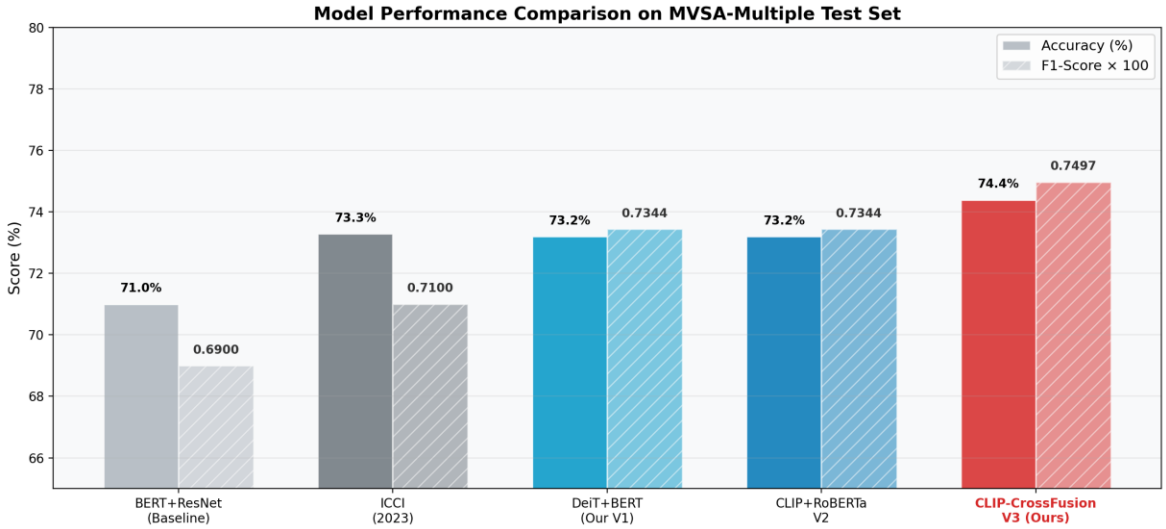


Fig. 4. Model performance comparison on the MVSA-Multiple test set. The proposed CLIP-CrossFusion Net with Focal Loss and class weighting achieves 74.39% accuracy.

TABLE III. PERFORMANCE COMPARISON ON THE MVSA-MULTIPLE TEST SET

Model	Acc. (%)	W-F1	Neg-F1	Neu-F1	Pos-F1
DeiT-Base + BERT (Concat)	58.76	0.5879	0.52	0.41	0.67
ViT-Base + BERT (Concat)	57.13	0.5669	0.49	0.39	0.65
Swin-Base + BERT (Concat)	56.63	0.5434	0.47	0.41	0.67
RoBERTa-L + CLIP ViT-L/14 (large-encoder)	71.13	0.7045	0.76	0.57	0.72
ICCI (2023) [14]	73.29	0.7100	–	–	–
CLIP ViT-B/32 + RoBERTa (cross-modal attention)	73.20	0.7344	0.84	0.59	0.75
CLIP-CrossFusion Net + Focal + CW (ours)	**74.39**	**0.7497**	**0.85**	**0.63**	**0.75**

B. MVSA-Single Results

Table IV presents results on MVSA-Single. The proposed ensemble of the base model, colour-contrastive variant, and RoBERTa-Large variant achieves 75.66% accuracy and 0.7517 weighted F1, outperforming DeiT+BERT (61.50%) and substantially improving over the single best model (74.12%). The Neutral F1 improves from 0.38 (base model) to 0.45 (ensemble), a significant gain on the hardest class. Fig. 5 visualises this comparison, showing the progressive accuracy gains from the base model to the final ensemble, with the best result (75.66%) approaching the ICCI benchmark of 79.33%.

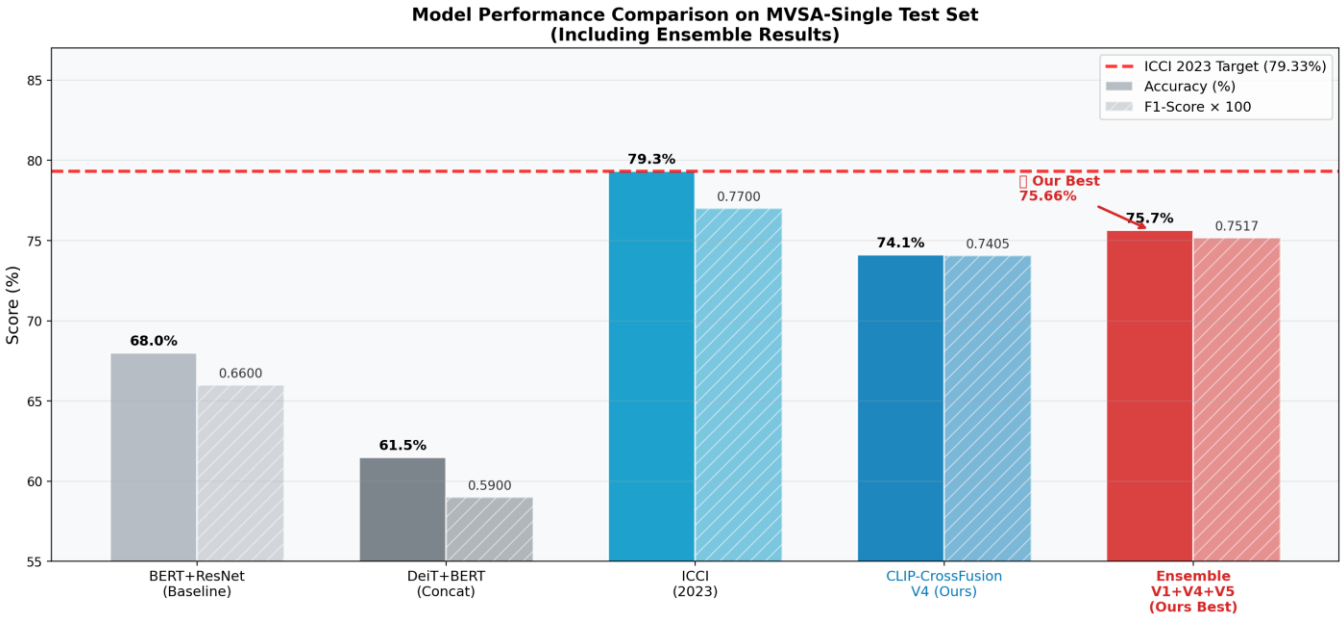


Fig. 5. Model performance comparison on the MVSA-Single test set.

TABLE IV. PERFORMANCE COMPARISON ON THE MVSA-SINGLE TEST SET

Model	Acc. (%)	W-F1	Neg-F1	Neu-F1	Pos-F1
BERT + ResNet (baseline)	68.00	0.6600	–	–	–
DeiT + BERT (Concat)	61.50	0.5900	–	–	–
ICCI (2023) [14]	79.33	0.7700	–	–	–
Base: CLIP + RoBERTa-Base	73.45	0.7288	0.65	0.38	0.83
+ Colour + Contrastive	74.12	0.7405	0.67	0.41	0.84
+ RoBERTa-Large	73.67	0.7355	0.67	0.40	0.83
Ensemble: base + colour-contrastive + large (ours)	**75.66**	**0.7517**	**0.67**	**0.45**	**0.85**

C. Ablation Study

Table V traces the contribution of each component across both datasets, showing systematic improvement as components are added.

TABLE V. ABLATION STUDY ACROSS BOTH DATASETS (X-ATTN = CROSS-MODAL ATTENTION, CW = CLASS WEIGHTS, CONT. = CONTRASTIVE LOSS)

Model	CLIP	X-Attn	Focal	CW	Colour	Cont.	Mult. (%)	Sing. (%)
DeiT + BERT	✗	✗	✗	✗	✗	✗	58.76	61.50
Base (CLIP + X-Attn)	✓	✓	✗	✗	✗	✗	71.13	73.45
+ Focal + CW	✓	✓	✓	✓	✗	✗	74.39	–
+ Colour + Contrastive	✓	✓	✓	✓	✓	✓	–	74.12
Ensemble (base + colour-contrastive + large)	✓	✓	✓	✓	✓	✓	–	**75.66**

D. Training Dynamics on MVSA-Multiple

Fig. 6 and Fig. 7 illustrate the training dynamics on MVSA-Multiple. Fig. 6 shows the overall accuracy and loss convergence, with the best checkpoint reached at epoch 10 (validation accuracy 74.71%, validation F1 0.7503). Fig. 7 tracks per-class F1 progression and demonstrates how targeted class weighting steadily improves the Neutral class F1 from 0.55 to 0.63 over training.

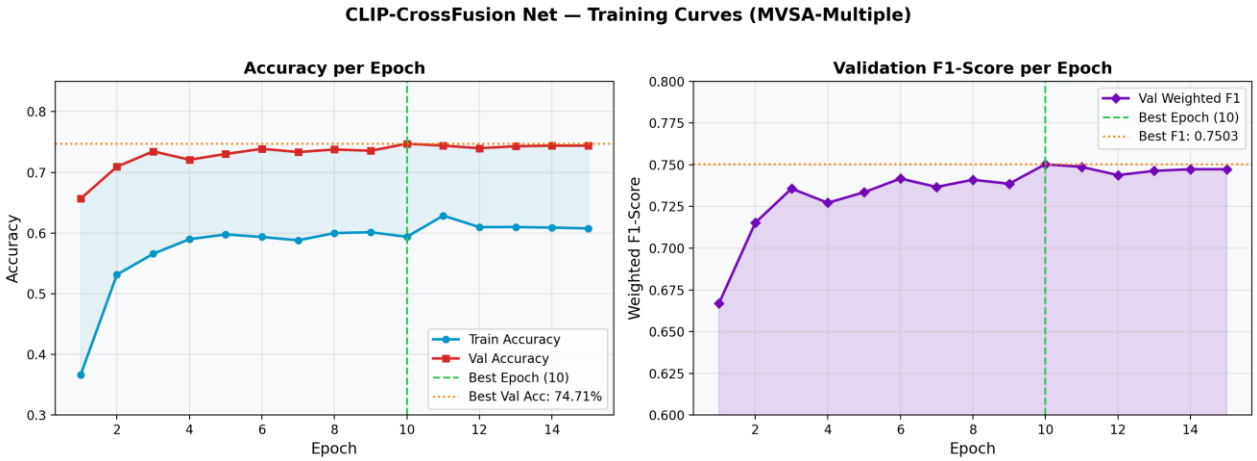


Fig. 6. Training curves on MVSA-Multiple. Best checkpoint at epoch 10 (Val. Acc.: 74.71%, Val. F1: 0.7503).

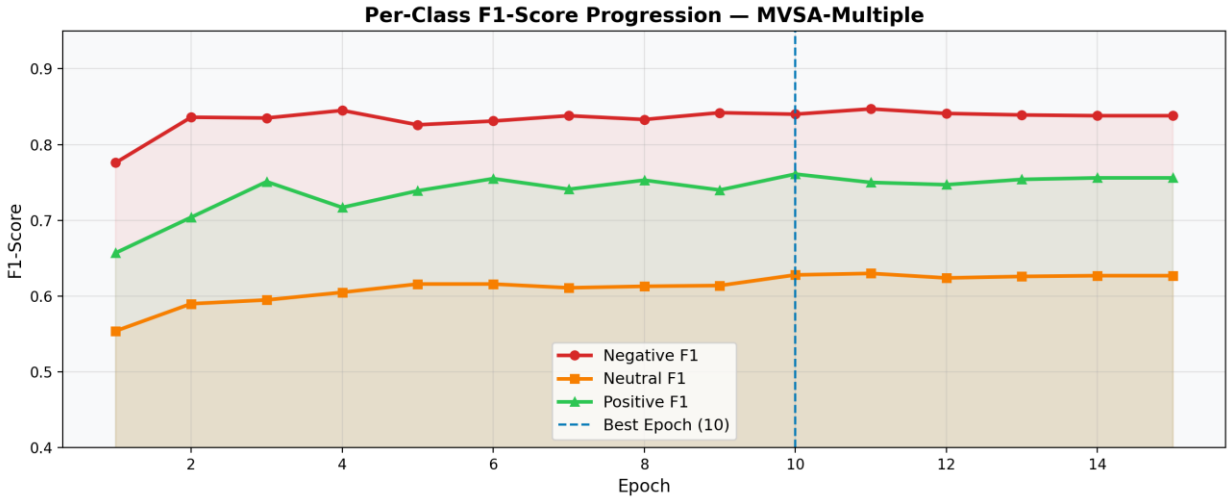


Fig. 7. Per-class F1 progression on MVSA-Multiple. The neutral class improves from 0.59 to 0.63 with targeted class weighting.

E. Training Dynamics on MVSA-Single

Fig. 8 and Fig. 9 present the training dynamics on MVSA-Single for the base model. Fig. 8 shows validation accuracy stabilising around epoch 17, with the best checkpoint at validation accuracy 73.61% and validation F1 0.7262. Fig. 9 tracks per-class F1 progression and highlights the delayed emergence of the Neutral class, which starts from 0.00 at epoch 1 due to the severe class imbalance of only 10.4% Neutral samples.

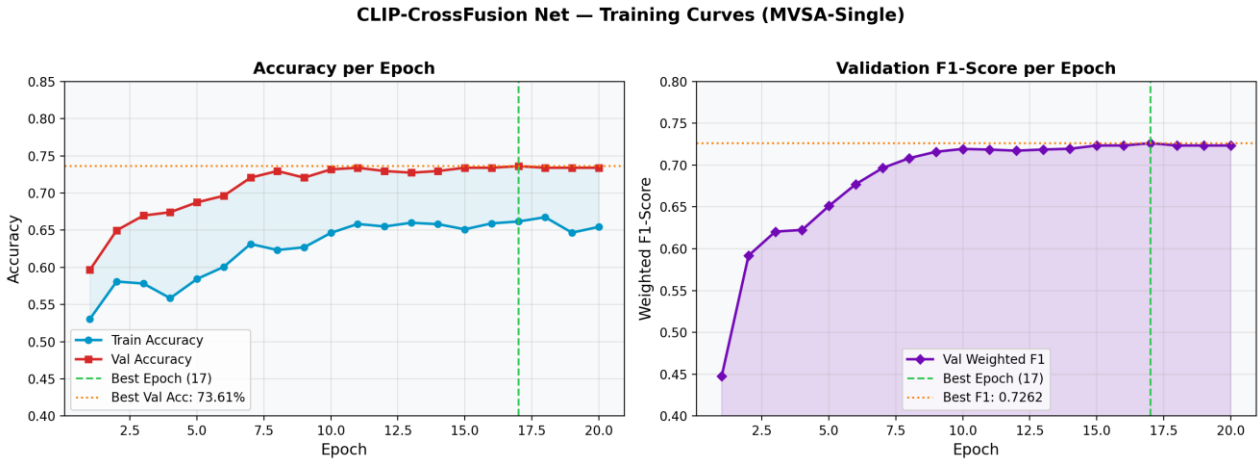


Fig. 8. Training curves on MVSA-Single (base model). Best checkpoint at epoch 17 (Val. Acc.: 73.61%, Val. F1: 0.7262).

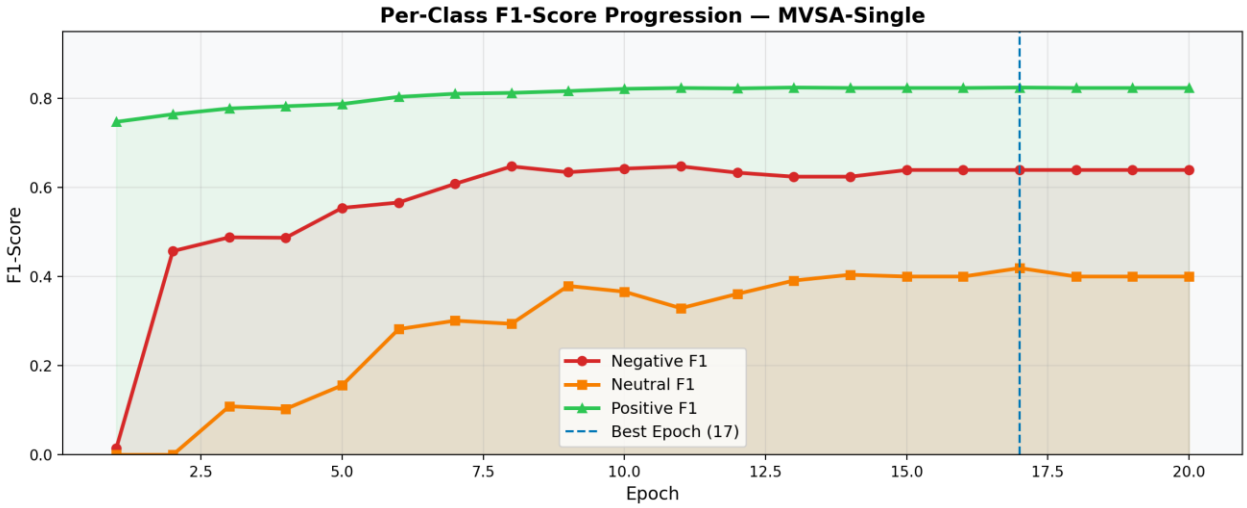


Fig. 9. Per-class F1 progression on MVSA-Single. The neutral class starts from 0 at epoch 1 due to severe class imbalance (10.4% of samples).

TABLE VI. DETAILED PER-CLASS RESULTS ON BOTH DATASETS (TEST SET WITH TTA)

Dataset / Model	Class	Prec.	Rec.	F1	Supp.	Observation
MVSA-Multiple, Focal + CW (74.39%)	Negative	0.91	0.80	0.85	893	Highest precision; clear negative signals
	Neutral	0.57	0.71	0.63	759	Hardest class; boosted by class weighting
	Positive	0.77	0.73	0.75	1,195	Largest class; good balance
MVSA-Single, ensemble (75.66%)	Negative	0.70	0.64	0.67	136	Stronger with ensemble than base alone
	Neutral	0.49	0.43	0.45	47	+0.07 over base alone; very small support
	Positive	0.82	0.87	0.85	269	Best among all versions

Table VI presents the per-class results on the MVSA-Single and MVSA-Multiple datasets. From Table VI, it is observed that, for the neutral class, only 47 samples are available as support.

F. Confusion Matrix Analysis

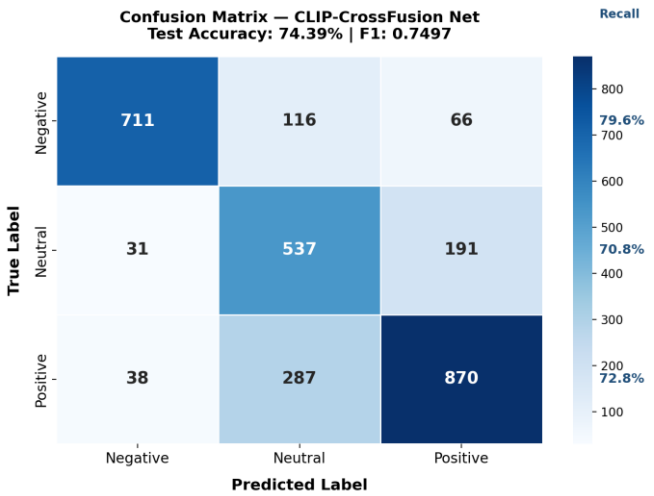


Fig. 10. Confusion matrix on the MVSA-Multiple test set (2,847 samples). Neutral to positive (191 cases) is the most frequent error.

Fig. 10 and Fig. 11 present confusion matrices for MVSA-Multiple and MVSA-Single, respectively. On MVSA-Multiple (Fig. 10, 2,847 samples), the most frequent error is Neutral samples being misclassified as Positive (191 cases), reflecting the contextual ambiguity of the Neutral class. On MVSA-Single (Fig. 11, 452 samples, base model), the Neutral class remains the hardest, with reliable estimation further limited by a test support of only 47 Neutral samples.

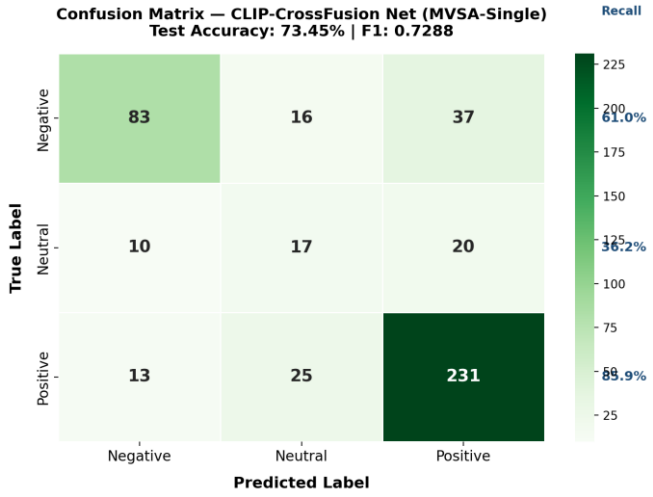


Fig. 11. Confusion matrix on the MVSA-Single test set (base model, 452 samples). Neutral is the most challenging class with only 47 test samples.

G. Cross-Dataset Comparison

Fig. 12 presents the final comparison across both datasets. The MVSA-Single ensemble slightly outperforms MVSA-

Multiple on overall accuracy (75.66% vs 74.39%) and Positive F1 (0.85 vs 0.75), while MVSA-Multiple is stronger on Negative F1 (0.85 vs 0.67) and Neutral F1 (0.63 vs 0.45). The Neutral gap reflects the much smaller Neutral support on MVSA-Single (10.4% vs 26.7%). MVSA-Multiple leads on Negative and Neutral F1 due to larger, more balanced class support.

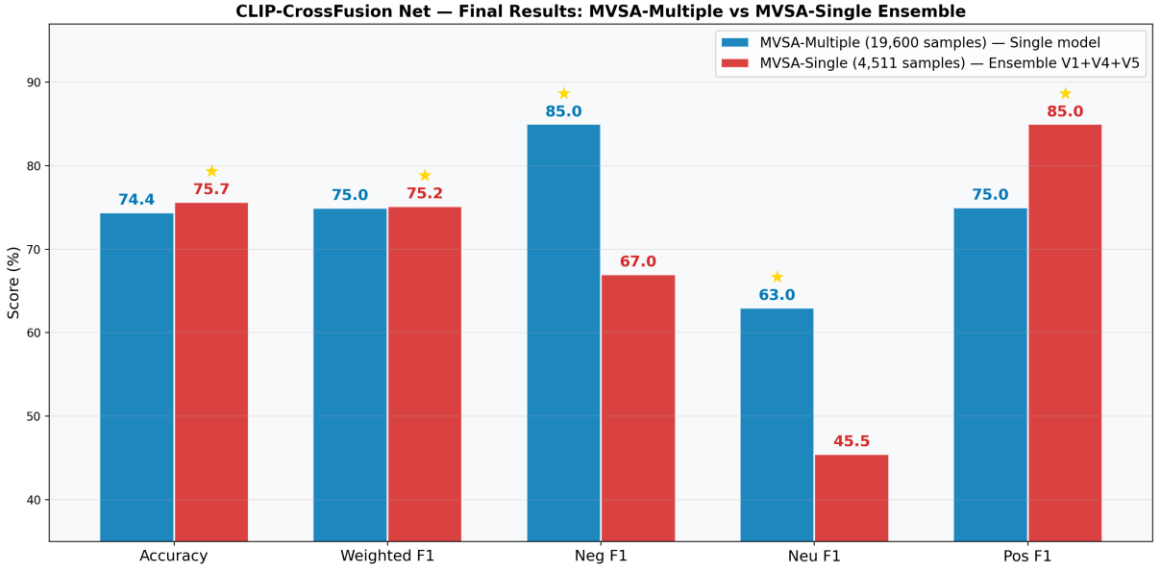


Fig. 12. Cross-dataset comparison of the final best models.

H. Comparison with State-of-the-Art

On MVSA-Multiple, the proposed model attains 74.39%, above ICCI (73.29%) and well above the concatenation baselines (56-58%). On MVSA-Single, the proposed ensemble reaches 75.66%, a 3.67 percentage-point gap from ICCI (79.33%). ICCI uses HSV colour features, which motivated the colour feature branch. The remaining gap likely comes from the richer colour modelling strategy of ICCI and different training data preprocessing. Biswas et al. [15] report 93.48% on MVSA-Multiple but use a highly specialised multi-stage pipeline with Mask-RCNN, AdaBoost, and Dempster-Shafer fusion that substantially increases complexity. The proposed end-to-end framework achieves competitive results with a simpler, more reproducible approach.

VII. Conclusion

This study presents CLIP-CrossFusion Net, a multimodal sentiment analysis framework combining CLIP ViT-B/32 and RoBERTa through bidirectional cross-modal attention for 3-class sentiment classification. The proposed model is evaluated comprehensively on both the MVSA-Multiple and MVSA-Single datasets with four model variants and a multi-model ensemble.

The main contributions and findings of the study are: 1) the proposed bidirectional cross-modal attention improves over simple concatenation by 12-15 percentage points on both datasets; 2) focal loss with class-balanced weighting pushes Neutral F1 from 0.59 to 0.63 on MVSA-Multiple and from 0.00 to 0.38 on MVSA-Single; 3) HSV colour features in the colour-contrastive variant add 0.67 percentage points of accuracy and improve Neutral F1 on MVSA-Single; 4) the multi-model ensemble achieves 75.66% on MVSA-Single, 2.21 percentage points over the single best model; and 5) both datasets reach above 74% accuracy on a single consumer GPU, demonstrating practical feasibility.

In future, the following works may be explored: region-based visual attention for fine-grained sentiment localisation; caption generation for image-to-text conversion as used by ICCI; multi-task learning with modality alignment objectives; larger vision-language models (CLIP ViT-L/14 with memory-efficient training); and extension to video-based multimodal sentiment analysis.

Declarations

Conflict of Interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Author Contributions. T S Kaveri and C K Roopa: conceptualization, methodology, investigation, writing - original draft; B S Harish and M S Kendagannaswamy: data curation, formal analysis, visualization; B S Harish: supervision, writing - review and editing.

Funding. The authors did not receive support from any organization for the submitted work.

Clinical Trial Registration. Not applicable. This study did not involve a clinical trial; therefore, no clinical trial registration number is available.

Consent to Publish. Not applicable.

Ethical Statement. This research did not involve human participants, animals, or clinical data. The study was conducted using publicly available datasets and computational machine learning methods. No ethical approval or informed consent was

required. The authors complied with all applicable institutional and journal ethical guidelines.

Consent to Participate. Not applicable; no human participants were involved.

Data Availability Statement. The datasets analysed during this study are publicly available from the repositories cited in the manuscript. All relevant dataset sources, accession details, and URLs are provided in the references and/or the dataset section (Section III). No new datasets were generated during this study.

REFERENCES

[1] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Information Fusion*, vol. 37, pp. 98-125, 2017.

[2] T. Niu, S. Zhu, L. Pang, and A. El Saddik, "Sentiment analysis on multi-view social data," in *MultiMedia Modeling*, Cham, Switzerland: Springer, 2016, pp. 15-27.

[3] P. D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 417-424.

[4] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014, pp. 1746-1751.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL-HLT)*, 2019, pp. 4171-4186.

[8] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[9] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: Harvesting opinions from the web," in *Proc. 13th Int. Conf. Multimodal Interfaces*, 2011, pp. 169-176.

[10] X. Yang, S. Feng, Y. Zhang, and D. Wang, "Multimodal sentiment detection based on multi-channel graph neural networks," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 328-339.

[11] T. Zhu *et al.*, "Multimodal sentiment analysis with image-text interaction network," *IEEE Trans. Multimedia*, vol. 25, pp. 3375-3385, 2023.

[12] Z. Li, B. Xu, C. Zhu, and T. Zhao, "CLMLF: A contrastive learning and multi-layer fusion method for multimodal sentiment detection," in *Findings Assoc. Comput. Linguistics: NAACL 2022*, 2022, pp. 2282-2294.

[13] G. Mu, Y. Chen, X. Li, L. Dai, and J. Dai, "Semantic enhancement and cross-modal interaction fusion for sentiment analysis in social media," *PLOS ONE*, vol. 20, no. 4, p. e0321011, 2025.

[14] J. An and W. M. N. W. Zainon, "Integrating color cues to improve multimodal sentiment analysis in social media," *Eng. Appl. Artif. Intell.*, vol. 126, p. 106874, 2023.

[15] S. Biswas, K. Young, and J. Griffith, "Exploring multimodal features for sentiment classification of social media data," Singapore: Springer, 2024, pp. 527-537, doi: 10.1007/978-981-99-8324-7_44.

[16] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748-8763.

[17] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[18] H. Touvron *et al.*, "Training data-efficient image transformers and distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347-10357.

[19] Z. Zhang, W. Wang, L. Zhu, and Z. Tang, "TAG-fusion: Two-stage attention guided multi-modal fusion network for semantic segmentation," *Digit. Signal Process.*, vol. 156, p. 104807, 2025.

[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

Artificial Intelligence for Rheumatoid Arthritis and Osteoporosis Diagnosis from Radiography, Computed Tomography, and Multimodal Clinical Data: A Systematic Review

Moulay Youssef ICHAHANE[1]*, Mohamed LACHGAR[2], Abdelouafi IKIDID[3],
Ahmed DOURHRI[4], Mohamed EL GHAZOUANI[5], Noureddine ASSAD[6]
Chouaîb Doukkali University, LTI/SI / CED Technology and Engineering, ENSA, El Jadida, Morocco[1, 4, 6]
Cadi Ayyad University, Higher Normal School (ENS), Computer Science Department, L2IS Laboratory, Marrakech, Morocco[2]
Cadi Ayyad University, UCA, Faculty of Sciences Semlalia, LCSS, Marrakesh, Morocco[3]
Chouaîb Doukkali University, ESIM, Polydisciplinary Faculty of Sidi Bennour, El Jadida, Morocco[5]

***Abstract*—Artificial intelligence (AI) is increasingly used to support musculoskeletal diagnosis from routine imaging, yet evidence remains fragmented across diseases, modalities, tasks, and validation designs. This systematic review synthesizes AI-based diagnosis, screening, grading, scoring, and structural assessment of rheumatoid arthritis (RA) and osteoporosis/osteopenia using radiography, computed tomography (CT), ultrasound, DXA-related imaging, and multimodal clinical data. Following PRISMA 2020 principles, Scopus, PubMed, Web of Science Core Collection, IEEE Xplore, and the ACM Digital Library were searched. Across the final exports, 2,239 records were identified; 247 full-text reports were assessed for eligibility, and 233 studies were included, while 25 reports could not be retrieved. CNN-based and transfer-learning approaches remained dominant, with increasing use of attention mechanisms, transformers, ensembles, explainable AI, and multimodal fusion. Osteoporosis research focused mainly on opportunistic screening and BMD-related prediction, whereas RA studies more often addressed diagnosis, joint-level damage, severity grading, and automated scoring. Methodological maturity remained uneven: 141 of 233 studies (60.5%) relied on internal validation only, whereas 44 (18.9%) reported external or multicenter validation. Future progress therefore depends less on marginal gains in headline metrics than on reproducible study design, robust external validation, calibration, and clinically integrated multimodal decision support.**

***Keywords*—Rheumatoid arthritis; osteoporosis; osteopenia; radiography; X-ray; CT; multimodal learning; deep learning; convolutional neural networks; transfer learning; systematic review**

I. Introduction

Artificial intelligence has become an increasingly important component of musculoskeletal and rheumatologic imaging because it can detect subtle image patterns, automate repetitive measurements, and support decision-making in settings where conventional interpretation is time-consuming or operator-dependent. The clinical rationale is especially strong in rheumatoid arthritis (RA), where structural joint damage requires reproducible detection and longitudinal assessment, and in osteoporosis/osteopenia, where underdiagnosis motivates opportunistic screening from routinely acquired images. Recent reviews confirm substantial growth of AI in both disease domains, but most syntheses remain disease-, modality-, or task-specific [30]–[35].

The technical landscape is correspondingly heterogeneous. Published systems range from binary disease detection to multiclass grading, bone-mineral-density (BMD) or T-score estimation, joint localization, erosion assessment, modified Sharp/van der Heijde scoring, and image-plus-clinical decision support. Conventional convolutional neural networks (CNNs) and transfer learning remain widely used, while attention mechanisms, vision transformers, ensemble learning, explainability, uncertainty estimation, and multimodal fusion are increasingly represented. The EHR-plus-imaging framework reported by Ichahane et al. [8] and the attention-enhanced ensemble for osteoporosis detection by Ichahane and Assad [28] illustrate this movement from isolated image classifiers toward richer patient-level and ensemble reasoning systems.

Despite rapid publication growth, the evidence is difficult to interpret as a coherent basis for clinical translation. Apparent performance depends on dataset scale and provenance, disease prevalence, task definition, reference-standard construction, patient-level data partitioning, and whether evaluation is internal, independently held out, or genuinely external. Existing meta-analyses have demonstrated promising diagnostic accuracy in selected osteoporosis settings [31]–[35], but such pooled estimates do not resolve broader cross-task heterogeneity, while RA-focused reviews cover a wider clinical and data spectrum [30]. Consequently, headline accuracy or AUC values can overstate maturity when validation design, calibration, leakage risk, and transportability are not considered jointly.

The central research gap is therefore not a lack of high-performing AI models, but the absence of a cross-disease synthesis that jointly evaluates clinical task, imaging modality, reference-standard quality, data provenance, risk of bias, and validation maturity in rheumatoid arthritis and osteoporosis/osteopenia. This review addresses that gap through

*Corresponding author.

a common analytical framework that 1) maps clinical objectives and task formulations; 2) compares imaging modalities, datasets, model families, and evaluation metrics; 3) evaluates reference-standard heterogeneity, internal versus external validation, and adapted QUADAS-2 risk-of-bias domains; 4) contrasts the methodological trajectories of the two disease areas; and 5) identifies barriers to clinical translation through an evidence-maturity perspective. By combining five bibliographic databases with article-level eligibility assessment and structured methodological appraisal, the review is designed to distinguish algorithmic performance from the strength and transportability of the supporting evidence.

The remainder of this study is organized as follows. Section II situates the present work within existing systematic reviews and meta-analyses. Section III describes the search strategy, eligibility criteria, PRISMA-based study selection, data extraction, risk-of-bias appraisal, and evidence-synthesis framework. Section IV presents the main results across the predefined research questions. Section V discusses evidence maturity, methodological limitations, and priorities for clinical translation, while Section VI concludes the study.

II. Related Works

A. Existing Review Landscape and Comparative Positioning

Existing reviews confirm the rapid growth of AI in both rheumatoid arthritis (RA) and osteoporosis, but their scope is usually disease-specific, modality-specific, or focused on a narrower clinical objective. For RA, Momtazmanesh et al. provided a broad state-of-the-art review spanning imaging, omics, EHRs, sensors, treatment response, and prognosis [30], whereas imaging-focused reviews have generally concentrated on selected arthritis tasks rather than a common diagnostic-imaging framework. For osteoporosis, recent systematic reviews and meta-analyses have addressed plain radiographs [31], panoramic dental radiography [32], CT-based deep learning [33], multi-imaging machine-learning diagnosis [34], and opportunistic CT screening [35]. These reviews demonstrate promising diagnostic performance but repeatedly identify substantial heterogeneity, limited external validation, and inconsistent reporting as barriers to translation.

B. Limitations of Existing Reviews

The prior literature nevertheless leaves three gaps relevant to the present review. First, RA and osteoporosis/osteopenia are rarely assessed within a single methodological framework, despite their shared reliance on image-based classification, quantitative scoring, transfer learning, and clinically meaningful reference standards. Second, prior syntheses often narrow eligibility to one modality or one task, which improves statistical comparability but limits cross-modality and cross-disease assessment of evidence maturity. Third, although several osteoporosis meta-analyses appropriately use QUADAS-2 [31]–[35], the broader diagnostic-AI literature remains inconsistent in reporting patient-level partitioning, leakage control, calibration, dataset provenance, and genuinely external validation. A recent IJACSA systematic review also illustrates the value of explicit search, eligibility, and comparative-synthesis structures for reproducible review reporting [9].

C. Added Value of the Present Review

The present review therefore extends the existing review landscape rather than duplicating a disease- or modality-specific meta-analysis. It jointly evaluates RA and osteoporosis/osteopenia, covers five bibliographic databases, and applies a common PRISMA-based workflow to 233 included primary journal studies (Table IV). The synthesis combines adapted QUADAS-2 domains with AI-specific appraisal of patient-level splitting, train-test leakage, validation design, data provenance, calibration, and interpretability. Because the included tasks range from binary screening to BMD regression, joint-level damage analysis, severity grading, and structured RA scoring, the review prioritizes structured narrative synthesis and evidence-maturity comparison rather than a single pooled diagnostic estimate.

This cross-disease perspective is particularly useful for separating architectural performance from evidence strength: an internally validated high-AUC classifier is not treated as equivalent to a multicenter or external-validation study. The review thus focuses on transportability, reference-standard quality, and clinical task complexity in addition to headline performance.

Comparative positioning against prior reviews. In osteoporosis, Yen et al. synthesized six plain-radiograph studies with QUADAS-2 and diagnostic meta-analysis [31]; Tarighatnia et al. focused on seven panoramic-radiography studies and additionally evaluated publication bias [32]; Wang et al. synthesized 24 CT-based deep-learning studies [33]; Zhao et al. expanded the scope to 60 X-ray, CT, and MRI studies [34]; and Ghareh Mohammadi and Sebro examined 31 opportunistic CT studies covering osteoporosis and spine-fracture screening [35]. These reviews provide stronger within-modality statistical comparability, but they do not address RA within the same framework.

For RA, the broad review by Momtazmanesh et al. spans imaging, omics, EHRs, sensors, treatment response, and prognosis [30], but was not designed as a diagnostic-imaging systematic review with QUADAS-style appraisal. The present review therefore complements rather than replaces these prior syntheses: it sacrifices a single pooled effect in favor of cross-disease comparison of clinical task, imaging modality, reference standard, data provenance, validation rigor, and AI-specific leakage risk across a substantially larger body of included primary studies.

III. Methods

This systematic review was conducted and reported according to PRISMA 2020 principles [10]. The review workflow comprised seven predefined components: review questions, information sources and search strategy, eligibility criteria, deduplication and staged study selection, structured data extraction, adapted risk-of-bias appraisal, and cross-dimensional evidence synthesis. Scopus was used as the primary discovery source, complemented by PubMed, Web of Science Core Collection, IEEE Xplore, and the ACM Digital Library. At each stage, study-selection decisions and reasons for exclusion were documented so that screening, retrieval, eligibility, and inclusion counts could be reconciled reproducibly.

A. Research Questions and Review Framework

Clinical AI for rheumatoid arthritis and osteoporosis/osteopenia can be organized along seven complementary dimensions: clinical objective, model family, imaging and data source, evaluation strategy, workflow role, task formulation, and methodological constraints on translation. These dimensions provide the analytical structure used to compare heterogeneous diagnostic-AI studies; their operational formulation is presented in Table I.

TABLE I. RESEARCH QUESTIONS AND ANALYTICAL FOCUS

RQ	Research question / analytical focus
RQ1	Clinical objectives targeted by AI systems.
RQ2	AI model families and methodological evolution.
RQ3	Imaging modalities, datasets, and data provenance.
RQ4	Evaluation metrics, reference standards, and validation rigor.
RQ5	Role of AI in diagnostic and reader-assistance workflows.
RQ6	Task formulations and label granularity.
RQ7	Methodological limitations and priorities for clinical translation.

These questions follow a diagnostic-review logic linking the target condition, AI index test, imaging input, reference standard, and clinically relevant outcome.

B. Search Strategy

The literature search combined four concept blocks: target condition (rheumatoid arthritis, osteoporosis, osteopenia), artificial-intelligence methods, medical-imaging inputs, and diagnostic or assessment tasks. The original Scopus search was generated on April 9, 2026, and the search was subsequently extended to PubMed, Web of Science Core Collection, IEEE Xplore, and the ACM Digital Library using database-specific adaptations of the same conceptual structure. The Scopus formulation is reported below to make the search logic immediately visible; the complete database-specific strategies are provided in Supplementary Table S2 and are publicly available through the Zenodo repository at https://doi.org/10.5281/zenodo.22751177. Exact execution dates for the supplementary database searches were not separately preserved in the frozen export metadata and are therefore not assigned retrospectively.

Research query:

TITLE-ABS-KEY (("rheumatoid arthritis" OR osteoporosis OR osteopenia) AND ("deep learning" OR "convolutional neural network*" OR CNN OR "ensemble learning" OR "attention mechanism*" OR "case-based reasoning" OR multimodal) AND ("x-ray" OR radiograph* OR "medical imag*" OR EHR OR "electronic health record*") AND (diagnos* OR detect* OR classif* OR predict* OR screening))

Applying the conceptually equivalent searches across the five sources yielded 1,053 Scopus records, 298 PubMed records, 551 Web of Science Core Collection records, 197 IEEE Xplore records, and 140 ACM Digital Library records (n = 2,239). Source overlap was handled by cross-database deduplication; therefore, database-specific retrieval counts are reported for transparency and are not interpreted as independent study contributions.

The resulting records were merged into a single screening dataset before DOI- and title-based deduplication.

C. Inclusion and Exclusion Criteria

Eligibility was defined by three core requirements: primary peer-reviewed evidence, medical imaging as an input to the AI system, and quantitative evaluation of a direct rheumatoid arthritis or osteoporosis/osteopenia diagnostic or assessment task. Exclusion rules removed non-primary publications, non-target conditions or outcomes, and image-processing studies without a diagnostic or scoring endpoint. The operational inclusion and exclusion criteria are given in Table II.

TABLE II. ELIGIBILITY CRITERIA

Code	Operational criterion
IC1	Primary journal study evaluating AI for direct RA or osteoporosis/osteopenia diagnosis, screening, grading, scoring, or structural assessment.
IC2	Medical imaging is an AI input, alone or with clinical data; no imaging input fails IC2.
IC3	At least one quantitative evaluation metric is reported.
EC1	Non-primary publication (conference/proceedings, review, editorial, letter/note, erratum, or book chapter).
EC2	Prognosis, epidemiology, fracture-only, treatment-response, or risk-factor study without direct target-condition assessment.
EC3	Image processing only (segmentation, preprocessing, enhancement, or compression) without a diagnostic/scoring endpoint.
EC4	Non-target disease (e.g., osteoarthritis); generic “arthritis” is not treated as RA.

D. Screening Process and PRISMA Flow

Records from all five sources were merged into a common screening dataset. Deduplication was performed first by normalized DOI and subsequently by normalized-title comparison when DOI information was missing or inconsistent. From 2,239 records, 892 duplicates were removed, leaving 1,347 unique records for screening. After pre-retrieval exclusions, 272 reports were sought for retrieval. Twenty-five reports could not be retrieved; 247 reports were assessed for eligibility in full text, 14 were excluded after full-text assessment, and 233 studies were included. The complete selection pathway is shown in Fig. 1.

Study selection followed conservative disease and evidence rules. Generic “arthritis” was not treated as rheumatoid arthritis unless an explicit RA target was demonstrated, and fracture-only studies were not considered osteoporosis evidence unless osteoporosis/osteopenia itself was directly evaluated. A report was considered retrieved only when the complete article had actually been obtained and assessed. Abstracts, previews, and metadata could support exclusion only when an exclusion criterion was already explicit; otherwise, potentially eligible reports remained unresolved until full-text retrieval was attempted. After targeted retrieval, 25 reports could not be retrieved and therefore were not assessed for eligibility. Of the 247 reports assessed in full text, 14 were excluded, and 233 studies were included in the review. The complete study-selection pathway and reconciled counts are shown in Fig. 1.

The 14 reports excluded after full-text assessment and the corresponding reasons for exclusion are detailed in Supplementary Table S3, available at https://doi.org/10.5281/zenodo.22751177.

Methodological safeguards. Four rules were enforced throughout the review: 1) a report was considered retrieved only after the complete article had been obtained and assessed; 2) no report was excluded solely because it was difficult to access; 3) generic arthritis was not interpreted as RA, and fracture-only outcomes were not interpreted as osteoporosis unless the target condition itself was directly assessed; and 4) raw accuracy or AUC values were not ranked across studies without considering task definition, reference standard, data provenance, and validation design.

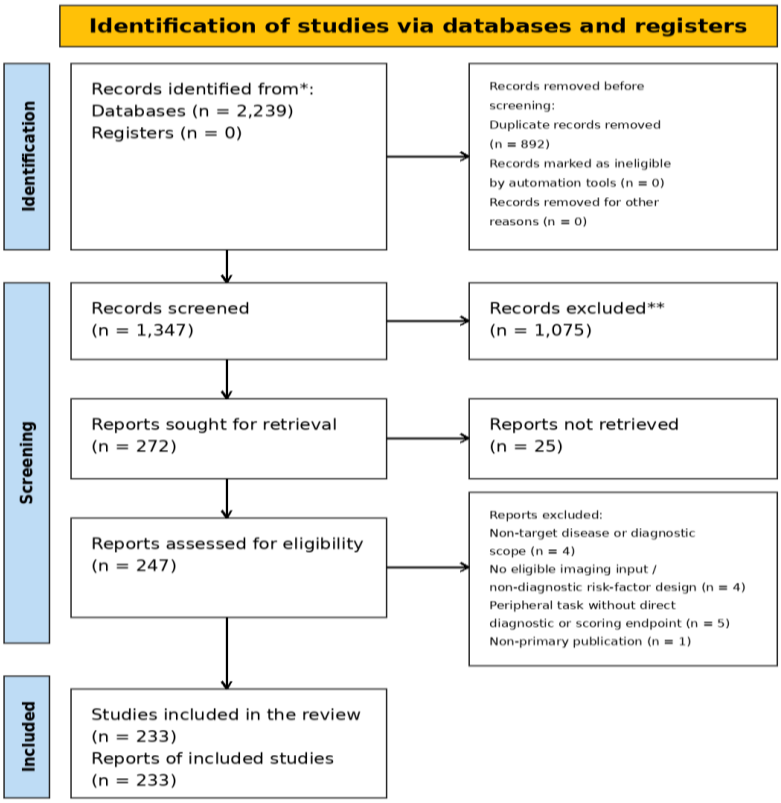


Fig. 1. PRISMA 2020 flow diagram of study identification, screening, retrieval, eligibility assessment, and inclusion.

E. Data Extraction and Synthesis Strategy

After study selection, full-text data extraction was performed using a standardized framework so that every included study could be compared on clinically and methodologically relevant dimensions. Extracted fields included publication year, disease focus, clinical objective, anatomical site, imaging modality, dataset size and provenance, AI architecture, multimodal inputs, task formulation, reference standard, validation design, external-validation status, and reported metrics (e.g., accuracy, sensitivity, specificity, F1-score, AUC, ICC, RMSE, and MAE). Patient-level splitting, potential leakage, class imbalance, calibration, interpretability, and data provenance were also recorded when reported. Metrics were preserved in their original task context rather than normalized into a single performance score.

F. Risk of Bias and AI-Specific Quality Appraisal

Risk of bias was appraised using an adapted QUADAS-2 structure covering patient selection, index test, reference standard, and flow/timing [29]. AI-specific extensions captured patient-level versus image-level splitting, train-test leakage risk, class imbalance, transparency of cohort construction, calibration, explainability, and independent or multicenter external validation. Domain judgments were retained separately; they were not collapsed into a single numerical "quality score," because such averaging can obscure distinct sources of diagnostic bias. The appraisal was used to contextualize performance and evidence maturity rather than as an automatic exclusion rule.

G. Evidence Synthesis and Evidence-Maturity Logic

A structured narrative synthesis was used because disease targets, anatomical sites, imaging modalities, class definitions, reference standards, dataset construction, model objectives, validation designs, and outcome metrics were too heterogeneous for a clinically defensible meta-analysis. Studies were synthesized by disease, clinical task, modality, model family, reference standard, and validation strategy. To avoid pseudo-precision, accuracy or AUC values from non-equivalent tasks were not pooled or averaged. A three-axis Evidence Maturity Framework was used to interpret the body of evidence across 1) clinical task complexity, 2) data richness/provenance, and 3) validation rigor. This framework is interpretive rather than a scoring system and is intended to distinguish high internal performance from stronger evidence of transportability and

clinical readiness. A single pooled effect estimate was therefore not pursued, because pooling non-equivalent targets, reference standards, imaging modalities, and validation designs would violate the clinical and statistical comparability required for a defensible diagnostic meta-analysis.

1) Evidence-maturity framework — clinical task. The continuum ranges from binary recognition to multiclass grading or BMD regression, and then to structured scoring, reader assistance, or clinically integrated decision support.

2) Evidence-maturity framework — data. The continuum ranges from small private single-center image sets to larger institutional/public cohorts and, at the highest level, multicenter or external cohorts combined with clinically meaningful structured data.

3) Evidence-maturity framework — validation. The continuum ranges from internal split or cross-validation to independent testing and, at the highest level, external/multicenter validation, calibration, reader studies, or prospective evaluation.

IV. Results

A. Characteristics of the Included Studies

Fig. 2 shows the publication-year distribution of the 233 included studies and highlights the rapid expansion of the evidence base after 2019. Fig. 3 presents the distribution of imaging modalities, showing the predominance of radiography and computed tomography. Fig. 4 summarizes validation maturity by distinguishing internal validation, independent test sets, and external or multicenter evaluation.

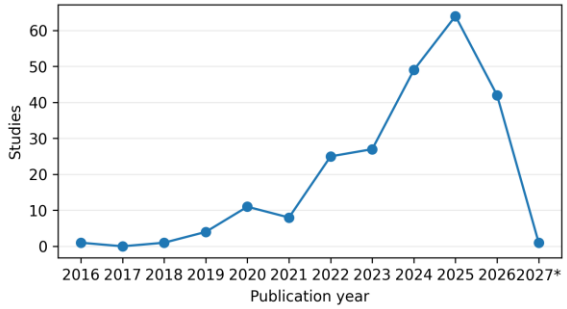


Fig. 2. Publication year of the 233 included studies. *The single study assigned to a 2027 issue had a 2026 DOI/online-publication record.

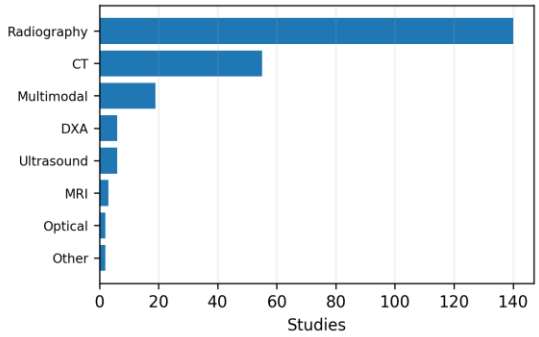


Fig. 3. Distribution of imaging modalities among the 233 included studies. Multimodal denotes studies using more than one imaging family.

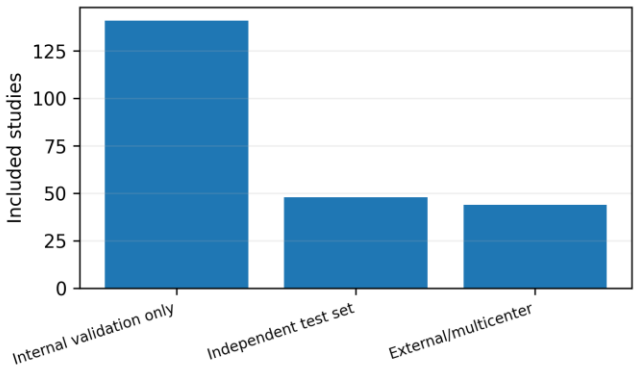


Fig. 4. Validation characteristics of the included studies: 141 internal-validation-only studies, 48 studies with an independent test set, and 44 studies with external or multicenter validation.

B. Overview of the Included Studies

Following deduplication, screening, report retrieval, and full-text eligibility assessment, 233 primary research studies published as journal articles were included: 198 addressing osteoporosis/osteopenia and 35 rheumatoid arthritis (Table V). Their target-condition distribution is shown in Fig. 5. The included evidence is methodologically diverse, but most studies still rely on internal validation; the corresponding risk-of-bias profile across adapted QUADAS-2 domains and the AI-specific split/leakage domain is shown in Fig. 6.

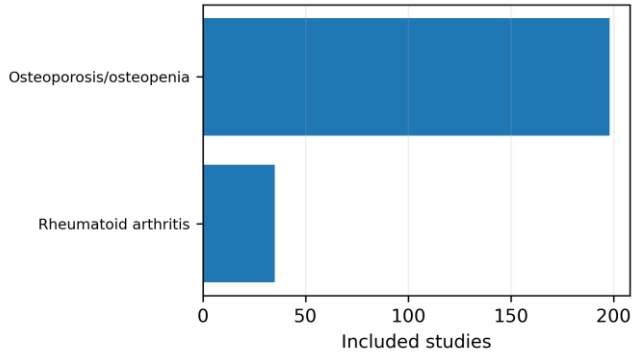


Fig. 5. Target conditions represented among the 233 included studies: 198 osteoporosis/osteopenia studies and 35 rheumatoid arthritis studies.

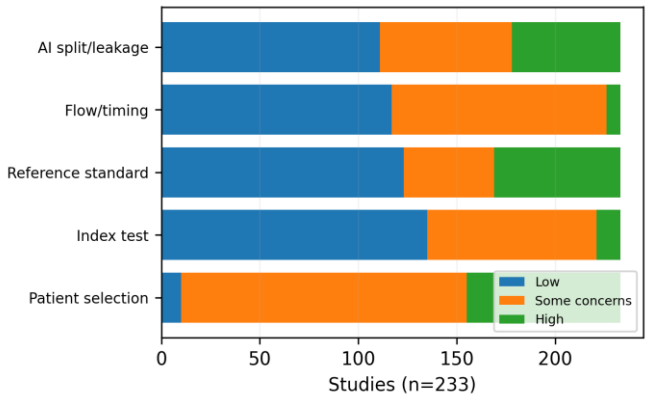


Fig. 6. Risk-of-bias assessment of the 233 included studies using adapted QUADAS-2 domains and an AI-specific split/leakage domain.

C. RQ1. Main Clinical Objectives Targeted by AI Systems

In osteoporosis and osteopenia, the dominant objective is opportunistic screening for low bone mineral density or disease using images acquired for other routine purposes [4]–[7], [12]–[14]. In RA, the objective space is more clinically granular and includes diagnosis, differential diagnosis, severity grading, joint-space or erosion analysis, and automated estimation of structured radiographic scores [2], [16], [17]. Overall, osteoporosis studies remain driven mainly by case finding and opportunistic screening, whereas RA studies are more deeply embedded in complex radiographic interpretation workflows.

D. RQ2. AI Model Families used Across the Included Studies

The dominant methodological pattern is the strong prevalence of deep convolutional image analysis. CNN-based and transfer-learning pipelines form the backbone of most osteoporosis studies and a large portion of RA studies. More recent work increasingly combines complementary architectures through attention or ensemble mechanisms; for example, the attention-enhanced E-DWE CNN ensemble proposed by Ichahane and Assad [28] integrates multiple pretrained backbones with adaptive weighting for knee-radiograph osteoporosis detection. In RA, multistage systems combining localization and classification are particularly prominent, while transformer-enhanced, radiomics-assisted, and explicitly interpretable systems are increasingly represented across both disease families. This methodological diversity is also illustrated by fine-tuned transfer learning for knee-radiograph osteoporosis classification [11], multimodal hand-radiograph/thermal RA modeling [21], interpretable hand-radiograph RA classification [22], and contextual deep-learning assessment of RA bone destruction [24].

E. RQ3. Data Types, Imaging Modalities, and Datasets

Plain radiography is the dominant study-level primary modality: 140/233 studies (60.1%). CT accounts for 55 (23.6%), multimodal imaging for 19 (8.2%), DXA-only and ultrasound for 6 each (2.6% each), MRI for 3 (1.3%), optical imaging for 2 (0.9%), and other imaging for 2 (0.9%). Osteoporosis research spans multiple anatomical sites and opportunistic imaging settings, whereas hand radiography remains central to RA. True image-plus-structured-data fusion remains comparatively uncommon. Additional included examples span cervical-radiography osteoporosis screening [12], spectral-CT osteoporosis assessment [23], few-shot knee-radiograph diagnosis [26], and custom deep-learning approaches for knee osteoporosis/osteopenia detection [27].

F. RQ4. Evaluation Metrics and Reported Performance

The included studies report a heterogeneous set of evaluation metrics. Accuracy, sensitivity, specificity, F1-score, and AUC dominate binary diagnostic tasks, whereas BMD estimation and RA scoring additionally use correlation, RMSE, MAE, and ICC. These metrics cannot be interpreted independently of validation design. A representative example is the hip-radiograph study by Jang et al. [4], in which AUC decreased from 0.867 on internal evaluation to 0.700 on an external cohort, illustrating the importance of transportability assessment.

Among the included studies, 141/233 (60.5%) used internal validation only, 48/233 (20.6%) reported an independent test set, and 44/233 (18.9%) reached external or multicenter validation. Consequently, the synthesis emphasizes validation rigor and reference-standard quality rather than ranking studies by headline accuracy.

G. RQ5. Impact of AI on Conventional Diagnostic and Reader-Assistance Workflows

In osteoporosis research, AI frequently turns routine images into opportunistic screening opportunities, allowing patients to be flagged for confirmatory DXA without a separate imaging pathway [5], [6], [13], [14]. In RA, AI is more often positioned as reader assistance, supporting joint localization, damage scoring, structured interpretation, and consistency [2], [17]. The evidence supports AI as a complementary decision-support layer rather than as a replacement for expert judgment.

H. RQ6. Dominant Task Formulations and Label Granularities

Osteoporosis studies are dominated by binary or near-binary screening tasks and BMD-related prediction [6], [7], [18], whereas RA studies are more heterogeneous and include binary recognition, RA-versus-OA discrimination, severity grading, joint-wise scoring, and end-to-end prediction of aggregate radiographic scores [2], [15]–[17]. Direct comparison of headline metrics is therefore misleading unless task complexity is considered.

I. RQ7. Limitations of the Current Literature and Opportunities for Future Work

Recurrent limitations include small private datasets, retrospective single-center designs, heterogeneous reference standards, insufficient reporting of patient-level splitting, limited calibration, and scarce external validation. The adapted risk-of-bias profile quantifies these concerns: 78/233 studies (33.5%) received a high-risk judgment for patient selection, 64/233 (27.5%) for the reference standard, and 55/233 (23.6%) for AI split/leakage. Only 44/233 studies (18.9%) reached external or multicenter validation. These weaknesses create risks of overfitting, spectrum bias, hidden data leakage, and site-specific performance inflation.

J. Cross-Dimensional Evidence Maturity

When the included studies are interpreted across task complexity, data richness, and validation rigor, a clear imbalance emerges. The field is strongest in algorithmic experimentation and internal discrimination performance, but substantially less mature in transportability and workflow-level evidence. Many studies occupy an intermediate position: they use clinically relevant imaging and modern architectures, yet remain dependent on retrospective single-center data and internal validation. The highest-maturity studies are those that combine clinically meaningful endpoints with independent or multicenter testing, transparent reference standards, and explicit attention to calibration, interpretability, or reader interaction. This pattern indicates that the next phase of progress is more likely to depend on study design and evidence quality than on marginal architectural novelty.

K. Summary of Key Findings

The consolidated evidence therefore reveals four consistent patterns. First, conventional radiography remains the most broadly deployed modality, while CT plays an important role in opportunistic osteoporosis screening and ultrasound remains particularly relevant to RA. Second, CNNs and transfer learning continue to dominate, although attention, transformers, ensemble strategies, and explainable AI are increasingly represented. Third, osteoporosis research is more frequently framed as opportunistic screening or BMD-related prediction, whereas RA research more often involves complex grading, localization, and scoring tasks. Fourth, validation quality remains the principal methodological bottleneck across both disease families, with external and prospective evaluation substantially less common than internal testing.

V. Discussion

A. From Algorithmic Performance to Evidence Maturity

The revised evidence base indicates that the central question is no longer whether AI can achieve high performance on selected RA or osteoporosis datasets, but whether that performance is supported by a sufficiently mature evidence design. Across the 233 included studies, 141 (60.5%) relied on internal validation only, whereas 44 (18.9%) provided external or multicenter evidence. This imbalance explains why high reported discrimination should not automatically be interpreted as clinical transportability.

B. Task Complexity, Data Richness, and Validation Trade-offs

The three-axis synthesis highlights an important trade-off. Simpler binary screening tasks can achieve high discrimination with relatively modest datasets, whereas structured RA scoring, multiclass grading, and multimodal decision support demand richer labels, stronger reference standards, and more robust validation. Similarly, increasing data richness does not automatically improve clinical credibility if data provenance, patient-level partitioning, or external transportability remain weak. This is why performance metrics must be interpreted jointly with task complexity and validation design rather than ranked in isolation. For osteoporosis, opportunistic screening remains the clearest near-term translational pathway. For RA, the strongest role is likely to be structured reader assistance, where AI supports localization, scoring consistency, and repetitive interpretation while preserving clinician oversight.

C. Persistent Validation and Reporting Gaps

The strongest recurring weakness is limited external validation. Only 44 of the 233 included studies (18.9%) reached external or multicenter evaluation, whereas 141 (60.5%) relied on internal validation only. High-risk judgments were assigned to 78 studies for patient selection, 64 for the reference standard, and 55 for AI split/leakage. These findings reinforce the need for multicenter evaluation, prespecified patient-level partitioning, transparent provenance, calibration, and reporting of clinically meaningful uncertainty.

D. Limitations of this Review

At the review level, restriction to primary journal articles improves comparability but excludes potentially informative conference research. Search syntax required database-specific adaptation, and the evidence base remains too heterogeneous for a valid pooled diagnostic-effect estimate. Twenty-five potentially eligible reports could not be retrieved despite targeted searches and therefore could not be assessed for eligibility; they are reported separately in the PRISMA flow as reports not retrieved. In addition, reporting quality in the source literature was uneven for patient-level splitting, calibration, reference-standard construction, and external validation, limiting the precision of cross-study risk-of-bias judgments. These constraints should be considered when interpreting completeness and generalizability. Because the review did not estimate a single common pooled diagnostic effect, formal funnel-plot or regression-based tests of publication bias were not methodologically appropriate across the full evidence base. Nevertheless, publication and selective-reporting bias remain plausible: AI studies with favorable discrimination may be more likely to reach journal publication, while negative, null, or poorly performing models may remain unpublished or incompletely reported. Restricting the synthesis to peer-reviewed journal articles may further accentuate this dissemination bias. These considerations reinforce the need to interpret high reported performance cautiously, particularly when external validation, calibration, or prespecified analysis protocols are absent.

E. Future Research and Translational Priorities

Future research should prioritize methodological and translational maturity over incremental architectural novelty. High-value priorities include multicenter external validation using geographically and technically diverse cohorts; prospective reader-assistance studies; standardized patient-level data partitioning and leakage controls; calibration and uncertainty reporting; explicit comparison against clinically meaningful baselines; transparent reference standards; and multimodal fusion that combines imaging with laboratory, demographic, and EHR information. Shared benchmarks and reporting checklists tailored to diagnostic AI would improve comparability, while reader studies and prospective workflow evaluation would clarify whether algorithmic gains translate into better clinical efficiency, consistency, and patient-level decision support.

F. Methodological Priorities for Clinical Translation

Methodological maturity in diagnostic AI depends on validation strategy, dataset scale and provenance, reference-standard consistency, patient-level partitioning and leakage control, calibration and uncertainty assessment, clinically grounded explainability, prospective workflow evaluation, and multimodal integration. Each recurring limitation therefore implies a corresponding study-design or reporting priority; these gap-to-action relationships are synthesized in Table III.

TABLE III. METHODOLOGICAL GAP-TO-ACTION MATRIX

Observed gap	Recommended reporting / study-design response
Limited external validation	Use prespecified independent or multicenter test cohorts; report transportability and subgroup performance.
Small or private datasets	Use larger shared or multicenter cohorts with transparent provenance and patient-level partitioning.
Heterogeneous reference standards	Report task-specific reference standards, label generation, adjudication, and image-reference timing.
Unclear splitting/leakage control	Partition at patient level; isolate preprocessing and augmentation; document leakage controls.
Sparse calibration/uncertainty	Report calibration, uncertainty, and decision analysis when clinically relevant.
Explainability without reader evidence	Link XAI to error analysis and reader-centered evaluation rather than plausibility alone.
Few prospective workflow studies	Use prospective reader or workflow studies with time, consistency, and decision-impact outcomes.
Limited multimodal context	Fuse imaging with clinical variables only when provenance and validation remain transparent.

VI. CONCLUSION

This systematic review synthesizes 233 primary research studies on AI-based diagnosis, screening, grading, scoring, and structural assessment of rheumatoid arthritis and osteoporosis/osteopenia. Across the included evidence, radiography remains the dominant imaging modality, while CNNs and transfer learning continue to underpin most systems and newer work increasingly incorporates attention, transformers, ensembles, explainability, and multimodal fusion.

The principal finding is methodological rather than architectural. Of the 233 included studies, 141 (60.5%) relied on internal validation only, whereas just 44 (18.9%) reported external or multicenter validation. Thus, high discrimination on retrospective or single-center datasets should not be equated with clinical transportability. The adapted risk-of-bias assessment further indicates persistent concerns related to patient selection, reference standards, and AI-specific split/leakage practices.

Future progress should therefore prioritize evidence maturity: multicenter and prospective evaluation, reproducible patient-level partitioning, transparent reference standards, calibration and uncertainty reporting, reader-interaction studies, and genuinely multimodal integration. For osteoporosis, opportunistic screening from routine imaging appears particularly promising; for RA, structured reader assistance and automated scoring are likely to be the most clinically relevant near-term applications.

DECLARATION ON GENERATIVE AI

Generative AI tools were used only for language refinement, document-formatting assistance, and consistency checking. All eligibility decisions, data extraction, methodological judgments, interpretation, and conclusions were made or critically verified by the authors, who retain full responsibility for the manuscript.

SUPPLEMENTARY MATERIALS

The supplementary materials associated with this systematic review are publicly available through the Zenodo repository at https://doi.org/10.5281/zenodo.22751177. Supplementary Table S1 provides the complete study-level extraction for the 233 included studies; Supplementary Table S2 provides the database-specific search strategies for Scopus, PubMed, Web of Science Core Collection, IEEE Xplore, and ACM Digital Library; and Supplementary Table S3 lists the reports excluded after full-text assessment together with the corresponding exclusion reasons.

REFERENCES

[1] G. S. Mate, A. K. Kureshi, and B. K. Singh, "An efficient CNN for hand X-ray classification of rheumatoid arthritis," Journal of Healthcare Engineering, vol. 2021, 2021.

[2] Z. Zhou et al., "RATING: Medical knowledge-guided rheumatoid arthritis assessment from multimodal ultrasound images via deep learning," Patterns, vol. 3, no. 10, 2022.

[3] Y. Peng, X. Huang, M. Gan, K. Zhang, and Y. Chen, "Radiograph-based rheumatoid arthritis diagnosis via convolutional neural network," BMC Medical Imaging, vol. 24, no. 1, 2024.

[4] R. Jang et al., "Prediction of osteoporosis from simple hip radiography using deep learning algorithm," Scientific Reports, vol. 11, no. 1, 2021.

[5] K. Fang, X. Zheng, X. Lin, and Z. Dai, "A comprehensive approach for osteoporosis detection through chest CT analysis and bone turnover markers: harnessing radiomics and deep learning techniques," Frontiers in Endocrinology, vol. 15, 2024.

[6] T.-Y. Yen et al., "Predicting osteoporosis from kidney-ureter-bladder radiographs utilizing deep convolutional neural networks," Bone, vol. 184, 2024.

[7] H. G. Nguyen et al., "Artificial intelligence system for predicting areal bone mineral density from plain X-rays," Osteoporosis International, vol. 36, no. 11, pp. 2167–2176, 2025.

[8] M. Y. Ichahane, N. Assad, and H. Ouahmane, "Enhancing rheumatoid arthritis diagnosis: Combining case-based reasoning on EHR data with deep learning on medical images," Journal of Universal Computer Science, vol. 32, no. 3, pp. 373–404, 2026.

[9] M. Aatila, M. Lachgar, H. Hrimech, and A. Kartit, "Machine learning techniques in keratoconus classification: A systematic review," International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, pp. 648–656, 2023.

[10] M. J. Page et al., "PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," BMJ, vol. 372, p. n160, 2021.

[11] U. B. Abubakar, M. M. Boukar, and S. Adeshina, "Evaluation of parameter fine-tuning with transfer learning for osteoporosis classification in knee radiograph," International Journal of Advanced Computer Science and Applications, vol. 13, no. 8, pp. 246–252, 2022.

[12] K. Tamai et al., "Deep learning algorithm for identifying osteopenia/osteoporosis using cervical radiography," Scientific Reports, vol. 15, no. 1, 2025.

[13] F. Gharehmohammadi and R. Sebro, "Deep learning opportunistic screening for osteoporosis and osteopenia using radiographs of the foot or ankle – a pilot study," European Journal of Radiology, vol. 184, 2025.

[14] B. S. S. Dias et al., "Osteoporosis screening: Leveraging EfficientNet with complete and cropped facial panoramic radiography imaging," Biomedical Signal Processing and Control, vol. 100, 2025.

[15] K. Ureten and H. H. Maras, "Automated classification of rheumatoid arthritis, osteoarthritis, and normal hand radiographs with deep learning methods," Journal of Digital Imaging, vol. 35, no. 2, pp. 193–199, 2022.

[16] H.-J. Wang et al., "Deep learning-based computer-aided diagnosis of rheumatoid arthritis with hand X-ray images conforming to modified total Sharp/van der Heijde score," Biomedicines, vol. 10, no. 6, 2022.

[17] H. Moradmand and L. Ren, "Multistage deep learning methods for automating radiographic Sharp score prediction in rheumatoid arthritis," Scientific Reports, vol. 15, no. 1, p. 3391, 2025. doi:10.1038/s41598-025-86073-0.

[18] S. Lee, E. K. Choe, H. Y. Kang, J. W. Yoon, and H. S. Kim, "The exploration of feature extraction and machine learning for predicting bone density from simple spine X-ray images in a Korean population," Skeletal Radiology, vol. 49, no. 4, pp. 613–618, 2020.

[19] S. Kumar, P. Goswami, and S. Batra, "Enriched diagnosis of osteoporosis using deep learning models," International Journal of Performability Engineering, vol. 19, no. 12, pp. 824–833, 2023, doi: 10.23940/ijpe.23.12.p7.824833.

[20] D. Çetintaş, G. Kılıçarslan, T. Tuncer, and D. Çetintaş, "Automated diagnosis of rheumatoid arthritis from hand radiographs using artificial intelligence: A retrospective study," Clinical Medicine Insights: Arthritis and Musculoskeletal Disorders, vol. 19, 2026.

[21] A. R. Kesavapillai, S. M. Aslam, S. Umapathy, and F. Almutairi, "RA-XTNet: A novel CNN model to predict rheumatoid arthritis from hand radiographs and thermal images," Diagnostics, vol. 14, no. 17, 2024.

[22] K. Cai et al., "Deep learning classification of rheumatoid arthritis in hand radiographs: interpretability insights and web application," ImmunoTargets and Therapy, vol. 14, pp. 1333–1345, 2025.

[23] H. Zhang et al., "Screening for osteoporosis based on IQon spectral CT virtual low monoenergetic images: Comparison with conventional 120 kVp images," Heliyon, vol. 9, no. 10, 2023.

[24] K. Miyama et al., "Deep learning-based automatic bone-destruction evaluation system using contextual information from other joints," Arthritis Research & Therapy, vol. 24, no. 1, 2022.

[25] Y.-J. Huang et al., "Radiographic bone texture analysis using deep learning models for early rheumatoid arthritis diagnosis," Journal of Imaging Informatics in Medicine, 2025.

[26] H. Xie et al., "A few-shot learning framework for the diagnosis of osteopenia and osteoporosis using knee X-ray images," Journal of International Medical Research, vol. 52, no. 9, 2024.

[27] S. M. Naguib, M. K. Saleh, H. M. Hamza, K. M. Hosny, and M. A. Kassem, "A new superfluity deep learning model for detecting knee osteoporosis and osteopenia in X-ray images," Scientific Reports, vol. 14, no. 1, 2024.

[28] M. Y. Ichahane and N. Assad, "Multi-Model Attention-Enhanced CNN Ensemble for Automated Osteoporosis Detection in Radiographic Knee Images," International Journal of Intelligent Engineering and Systems, vol. 18, no. 3, pp. 177–194, 2025, doi: 10.22266/ijies2025.0430.13.

[29] P. F. Whiting et al., "QUADAS-2: A revised tool for the quality assessment of diagnostic accuracy studies," Ann. Intern. Med., vol. 155, no. 8, pp. 529–536, 2011.

[30] S. Momtazmanesh, A. Nowroozi, and N. Rezaei, "Artificial Intelligence in Rheumatoid Arthritis: Current Status and Future Perspectives: A State-of-the-Art Review," Rheumatology and Therapy, vol. 9, pp. 1249–1304, 2022, doi: 10.1007/s40744-022-00475-4.

[31] T.-Y. Yen, C.-S. Ho, Y.-P. Chen, and Y.-C. Pei, "Diagnostic Accuracy of Deep Learning for the Prediction of Osteoporosis Using Plain X-rays: A Systematic Review and Meta-Analysis," Diagnostics, vol. 14, no. 2, art. 207, 2024, doi: 10.3390/diagnostics14020207.

[32] A. Tarighatnia, M. Amanzadeh, M. Hamedan, A. Mohammadnia, and N. D. Nader, "Deep learning-based evaluation of panoramic radiographs for osteoporosis screening: a systematic review and meta-analysis," BMC Medical Imaging, vol. 25, art. 86, 2025, doi: 10.1186/s12880-025-01626-z.

[33] A. Wang et al., "Deep Learning-Assisted Automated Diagnosis of Osteoporosis Based on Computed Tomography Scans: Systematic Review and Meta-Analysis," Journal of Medical Internet Research, vol. 27, e77155, 2025, doi: 10.2196/77155.

[34] R. Zhao et al., "The Diagnostic Value of Image-Based Machine Learning for Osteoporosis: Systematic Review and Meta-Analysis," Journal of Medical Internet Research, vol. 28, e75965, 2026, doi: 10.2196/75965.

[35] F. Ghareh Mohammadi and R. Sebro, "Artificial Intelligence for Opportunistic Screening for Osteoporosis and Spine Fractures Using Computed Tomography: A Systematic Review and Meta-Analysis," Journal of Computer Assisted Tomography, online ahead of print, 2026, doi: 10.1097/RCT.0000000000001899.

APPENDIX A

Tables IV and V present representative osteoporosis/osteopenia and rheumatoid arthritis studies, respectively, illustrating the range of imaging modalities, model families, performance measures, and validation designs discussed in Section IV. The complete study-level extraction for all 233 included studies is provided in Supplementary Table S1. The database-specific search strategies are provided in Supplementary Table S2, and the 14 reports excluded after full-text assessment with their exclusion reasons are listed in Supplementary Table S3. All supplementary files are publicly available through Zenodo at https://doi.org/10.5281/zenodo.22751177.

TABLE IV. REPRESENTATIVE INCLUDED OSTEOPOROSIS/OSTEOPENIA STUDIES

Study	Dataset / Modality	Model	Metrics / Best values	Validation / comparative insight
Lee et al. (2020) [18]	Spine X-ray; 334 participants	VGG feature extraction + RF	AUC 0.74; Acc. 0.71; Sens. 0.81; Spec. 0.60	Internal only; early proof-of-concept with moderate performance.
Jang et al. (2021) [4]	Hip radiography; 1001 images + external validation	VGG16 + nonlocal network	Internal AUC 0.867; External AUC 0.700	External validation; AUC decreased from 0.867 internally to 0.700 externally.
Kumar et al. (2023) [19]	Knee X-rays; 372 images	Eight ImageNet-pretrained transfer-learning models	Best average Acc. 86.36%	Internal train/test split; broad transfer-learning benchmark without external validation.
Fang et al. (2024) [5]	Chest CT + BMD; 488 patients	Radiomics + 2D/3D DL	Best AUC 0.906	Independent single-center test set; strong CT-based comparative study.
Yen et al. (2024) [6]	KUB radiographs + DXA; 8913 pairs	DeepDXA-KUB CNN	AUROC 0.939 lumbar; 0.947 hip	Internal validation only; large real-world opportunistic screening cohort.
Nguyen et al. (2025) [7]	7060 pelvis/spine radiographs	Ensemble of seven DL models	AUC 0.96–0.97	Independent within-cohort test set; large population-based BMD prediction study.
Gharehmohammadi & Sebro (2025) [13]	Foot/ankle radiographs; 907 patients	Customized DL	Acc. 94.65%; AUC 0.87	Internal patient-level split; opportunistic screening from routine imaging.
Dias et al. (2025) [14]	Panoramic radiography	EfficientNet B5/B6/B7	Development CV Acc. 95–99%; separate DXA-referenced cohort evaluated	Independent DXA-referenced test cohort after development CV; stronger than CV alone.

TABLE V. REPRESENTATIVE INCLUDED RHEUMATOID ARTHRITIS STUDIES

Study	Dataset / Modality	Model	Metrics / Best values	Validation / comparative insight
Üreten & Maraş (2022) [15]	Hand radiographs; held-out test	YOLOv4 + VGG-16	RA vs normal: Acc. 90.7%; Sens. 92.6%; Spec. 88.7%; AUC 0.97	Internal held-out test; single-center differential-diagnosis study.
Mate et al. (2021) [1]	Hand radiographs; 290 images	Efficient CNN	Acc. 94.46%	Internal held-out test; strong binary classification on a compact dataset.
Zhou et al. (2022) [2]	Multimodal ultrasound; prospective + external test sets	RATING knowledge-guided DL	Prospective Acc. 86.1%; External 85.0%	Prospective and external test sets; comparatively mature multimodal RA validation.
Peng et al. (2024) [3]	9714 hand radiograph exams	CNN	AUC 0.955 RA vs not RA	Patient-level internal train/test split; large multicenter-source development dataset but no independent external cohort.
Wang et al. (2022) [16]	240 train / 104 test hand radiographs	Five CNN architectures	Recognition AUC 97.80%	Internal test set; strong recognition performance, with staging/scoring remaining more complex.
Moradmand & Ren (2025) [17]	679 training patients + 291 external test subjects	UNet + YOLOv7 + ViT multistage	Joint ID Acc. 99%; RMSE 9.73; MAE 5.35; ICC 0.702	Independent unseen-patient test set; multistage automated global Sharp-score prediction.
Huang et al. (2025) [25]	891 early RA + 1237 non-RA	Deep-TEN vs ResNet-50	AUC 0.69–0.73	Internal train/validation/test split; highlights the difficulty of early-RA discrimination.
Cetintas et al. (2026) [20]	311 RA + 259 controls	Attention DenseNet121/169	Acc. 88%; Recall 91%	Internal single-center train/validation/test split; attention-based classification with moderate data volume.

A Hybrid System with Automatic Classification and RAG for Moroccan Court Rulings

Sara IBRIHICH[1], Ahmed OUSSOUS[2], Ouafaa IBRIHICH[3], Mustapha ESGHIR[4]
Faculty of Sciences, Mohammed V University in Rabat, Morocco[1]
Faculty of Sciences and Technologies of Mohammedia (FSTM), Hassan II University, Morocco[2]
Mohammed V University in Rabat, Morocco[3, 4]

***Abstract*—The Accessibility of Moroccan court judgments and decisions is significantly limited due to the lack of a specialized Arabic Judicial information retrieval architecture. Current NLP technologies focus on Modern Standard Arabic and fail to capture the specialized procedural language of judicial rulings from the courts, resulting in inadequate search and query answering assistance for legal professionals. This study introduces AraLex-RAG, a system that combines document extraction, automated categorization, and Retrieval-Augmented Generation (RAG) for Moroccan Arabic court texts. Neither the retrieval nor the classification components need a GPU; however, the RAG module has the possibility to use an external LLM API for generation. This study presents AraLex-RAG, a benchmark dataset of 2855 unstructured Moroccan judicial judgments in 26 different legal classes. It has been created using an automated workflow that combines specific Arabic text normalization, optical character recognition, and balanced class sampling. The information retrieval system is a combination of similarity-based semantic search and keyword-based search, and then uses Reciprocal Rank Fusion to fuse their results. The document categorization is based on a character-level text analysis approach mixed with a linear classifier, while the question-answering feature creates Arabic replies based on retrieved court rulings using a broad language structure. An accuracy of 92.87 percent is achieved (std 1.33 percent) and 76.15 percent macro F1 across 26 categories using five-fold stratified cross-validation on 2853 documents, outperforming the previous best on Moroccan court judgments (80.51 percent on a six-class subset) on a challenge 4 times tougher. Using 3,936 document query pairings, dense semantic retrieval performs better than the search-based keyword in all four assessment measures. On 45-stratified pairs of questions and answers, the query-answering component obtains a cosine similarity value of 0.845, which is similar to the most advanced Arabic query-answering systems on structured fields.**

Keywords*—Legal information retrieval; BM25; FAISS; Arabic NLP; TF-IDF; text classification; LinearSVC; Moroccan legal judgments

I. Introduction

Legal information retrieval (LIR) involves identifying pertinent statutes, law cases, and court rulings in answer to a legal search query [22]. Efficient information extraction systems significantly diminish the time of researching and enhance the uniformity of court judgments by guaranteeing that pertinent legal precedents are not ignored or overlooked. LIR systems have attained stability and maturity for the English language via platforms such as LexisNexis and Westlaw [21], as well as studies like COLIEE [1], and even Chinese language is starting to gain focus and interest [2]. On the other hand, the Arabic legal corpora lack similar architecture. This difference and void are most pronounced in Morocco, where court rulings are articulated in a specialized form of formal and official legal Arabic that draws on traditional structures, field-specific lexicon, and administrative norms, which are not well represented by broader-purpose Arabic NLP tools.

Arabic has significant NLP obstacles: its morphologically complex architecture generates several forms of words from the same root; diacritic structures that clarify context are often absent in written text; and right-to-left directional display imposes software limitations. In the Moroccan Judicial field, these difficulties are aggravated by the lack of any annotated legal corpus, the particularity of Moroccan procedure-based Arabic, and the extraction of most court judgments from paper archives that are not available in a machine-readable format. Researchers are unable to expand on previous studies or methodically assess methodologies since the field does not presently provide a standard for Moroccan court judgments, a retrieval mechanism, or a validated question-answering pipeline. This study concurrently covers all three different unexplored areas.

This study is organized as follows: 1) AraLex-RAG: One of the golden standard corpora for the Moroccan Arabic juridical information extraction system, consisting of 2855 unstructured court judgments and rulings spanning 26 legal areas. Without a common dataset, reliable studies on Moroccan judicial Artificial intelligence are unattainable; this corpus establishes the basis for any future endeavors in the field. 2) Prove that deep learning is not as effective as traditional approaches when it comes to limited resources for Arabic legal categorization. Our classifiers overcome the preceding best on Moroccan judicial judgments, which only tackled six categories, while LinearSVC and TF-IDF obtain state-of-the-art accuracy over 26 categories. 3) The quantified extraction assessment on Moroccan Arabic court rulings demonstrates that intensive semantic search consistently surpasses keyword-based retrieval in this field. Level category research indicates that the difficulty of retrieval is associated with file similarity within a class, a discovery that informs further corpora development. 4) An exhaustive query-responding system assessed on Arabic legal text, using a major language approach based on recovered court rulings. Outcomes are similar to leading-edge Arabic query-responding systems in structured

areas, indicating that outstanding retrieval systems may be attained using freely available open-source frameworks. 5) A comprehensive framework that integrates OCR, information retrieval, categorization, and query answering into a single program operable on normal hardware without the need for a GPU. AraLex-RAG is a full tool for Moroccan legal experts and specialists.

The subsequent sections of this work are structured as follows. Section II examines pertinent background and related studies on international juridical data extraction, natural language processing in Arabic legal, current Arabic legal baseline, hybrid extraction techniques, and extracted augmented generation for legal query answering. Section III outlines the dataset creation process, system design, and assessment procedure. Section IV defines quantitative outcomes for categorization, extraction, and query responding. Section V examines the results, limits, and real-world consequences for legal experts in Morocco. Section VI concludes and highlights avenues for further research.

II. Related Work

A. International Legal Information Retrieval Landscape

Judicial information retrieval has attained significant development in various highly resourced languages, establishing robust baselines and offering valuable insights for developing fields like Moroccan Arabic. The COLIEE sharing task sets [1] are the foremost English legal information retrieval reference, necessitating technologies to extract pertinent law cases and execute legal entailment logic. The highest-performing systems of 2023 integrate BM25 for preliminary candidate retrieval with using BERT-based neural re-ranking [1], illustrating the continued significance of hybrid linguistic semantic methodologies despite the prevalence of huge pre-trained systems. LEGAL-BERT [2], a specialized variant of BERT trained on judicial rulings, legislation, and legal agreements, demonstrates that pre-training language models on domain-specific legal content consistently outperforms generic-purpose systems. In a Recent study, Santosh and Upadhya [3] established a cross-lingual section extraction standard utilizing European court rulings on human rights and discovered that multilingual models that have been pretrained continually perform poorly compared to monolingual peers in judicial documents, underscoring that linguistic scope compromises field depth.

In French, Authors [4] developed a binary classification approach to identify the existence of a cost allocation clause referred to as Article 700 in 1994 decisions from the French court of appeals. By combining TF-IDF vectorization with conventional classifiers, they surpass the domain-specific Judicial CamemBERT transformer's accuracy of 0.96 on targeted judgment sections. This discovery is enlightening: for well-specified classification assignments on organized law data sets, frequency-based approaches may rival transformer-based model designs, a conclusion that directly and profoundly impacts the architecture of AraLex's categorization unit.

The CAIL2018 dataset [5] in China, including 2.6 million criminal case judgments, has established itself as the standard norm for charge prediction and sentencing quantification. Traditional SVM classification methods attain 94 % accuracy in charge prediction, surpassing most neural methodologies on this extensive, well balanced data. The translation of these results into Arabic, where corpora are much smaller and class borders are semantically and linguistically ambiguous, demands rigorous methodological adjustment.

B. Natural Language Processing in Moroccan and Arabic Legal Field

Zahir et al. [6] constructed an initial corpus of Court of Cassation judgments and trained CNN and GloVe embedding models on a six-category judgment prediction task, achieving an accuracy of 80.51%. The effort is authentic and beneficial, although the scope is limited: six general categories, just categorization, without data extraction or quality assurance. Word-level embedding techniques encounter difficulties with Arabic syntax when the training data is insufficient. Another study [7] indicates superior performance (94.58%, F1 = 94.45%) on a seven-class test using TF-IDF with SVM, specifically with legislative codes, documents from the government characterized by uniform formatting rather than court rulings. Court rulings are more complex and challenging. Neither research establishes a public baseline nor assesses extraction.

Arabic typology poses a further difficulty: extensive suffixation, prefixation, and cliticization result in significant surface form diversity, making vocabulary matching risky without exhaustive planning [20]. Systems pre-trained on Egyptian or Gulf Arabic have challenges when adapting to the Moroccan judicial records, which integrate French civil law terminology with Maliki jurisprudence in manners that generic embeddings fail to adequately represent.

C. Legal Arabic Standard: Moroccan Gap and Saudi Focus

The most advanced Arabic legal AI standards focus on Saudi law. ArabLegalEval [8] evaluates LLMs using multiple-choice based and open-ended inquiries, according on Saudi Ministry of Justice laws and Board of Experts norms. Even GPT-4o attains just around 77% on zero-shot multiple-choice question tasks, serving as a pertinent reminder that robust general approaches continue to have difficulties with domain-specific judicial Arabic. However, these benchmarks assess a model's pre-existing knowledge; they are not retrieval techniques and were not intended to locate content within a corpus. They also address Saudi law, which functions under a distinct legal system compared to Morocco. Moroccan law integrates French civil law with Maliki jurisprudence, resulting in a juridical Arabic that diverges from Gulf dialects in terminology, syntax, and procedural framework. There is no baseline or retrieval mechanism for Moroccan court rulings.

D. Reciprocal Rank Fusion and Hybrid Extraction

BM25 and dense retrieval provide distinct advantages, a fact that is well-known. BM25 is dependable for exact matching, particularly article numerals, party nomenclature, and procedural terminology. Dense retrieval, as shown by Karpukhin et al. [9] using Dense Passage Retrieval, identifies relevant documents even when the query and document do not share any vocabulary, by analyzing trained vector representations. In open-domain QA assessments, dense

retrieval demonstrates superior recall at shallow depths, although BM25 performs comparably when keyword matching is paramount.

Reciprocal Rank Fusion [10] integrates ranking lists by assigning a score of 1/(k + rank) to each document from each system and aggregating the scores. The constant k = 60 prevents the highest-ranked document from dominating when an extraction algorithm exhibits excessive confidence. RRF requires neither training data nor score normalization, offering practical benefits in low-resource environments. Arabic legal inquiries combine precise terminology (case IDs, article numbers) with more general concepts (legal norms, procedure classifications), making the hybrid method particularly suitable.

E. Generation of Legal Questions-Answering Using Retrieval-Augmented Services

Lewis et al. [11] formalized Retrieval-Augmented Generation: the model obtains relevant materials prior to formulating responses, rather than relying only on parametric memory. This is of greater significance in legal applications than in most other fields. A memory-based system may create a seemingly authoritative legal assertion without any supporting documentation [12]. A citation referencing the exact court ruling may be verified. English legal QA systems like LEGAL-BERT [2] demonstrate enhanced response fidelity when employing RAG compared to its absence.

The study by Al-Rasheed et al. [13] presents Arabic retrieval-augmented generation that assesses GPT-3.5, GPT-4, and other specific Arabic versions on a dictionary question-answering challenge. The extracted context is a dictionary entry: concise, well-structured, and immediately relevant. GPT-4 achieves a Token F1 score of 0.90 and a Cosine Similarity of 0.87. While it serves as a valuable reference, dictionary entries provide the most concise form of retrieval settings. Unstructured court judgments are extended, more convoluted, and need the approach to extract pertinent information concealed inside extensive legal text.

III. Methods and Materials

A. Dataset

1) Data collection: The AraLex-RAG corpus consists of 2855 Moroccan judicial rulings sourced from tribunal archives in both digital (DOCX, PDF) and scanned picture forms. The papers consist only of unstructured, informal Arabic text devoid of markup, annotations, or predefined categories. The document length ranges from 150 to more than 2,000 tokens, with an average of around 420 tokens. Each record is classified into one of 26 judicial classifications, including family, property, civil, administrative law, and labour, as determined by the tribunal department code of the document.

2) Preprocessing pipeline: All documents undergo a uniform four-phase (Fig. 1) pipeline, implemented consistently throughout both corpus construction and query execution to guarantee uniformity.

Phase 1 OCR: Scanned picture documents are transformed into text through the application of Tesseract with the Arabic language model (ara).

Phase 2 Quality Filtering: Records producing fewer than 30 characters post-OCR extraction are eliminated as retrieval failures.

Phase 3 Arabic normalization: a regex-based pipeline eliminates diacritics (tashkeel), converts all Alef variations (,إ ,أ آ ,ٱ) to the plain Alef (ا), and removes non-Arabic letters and extraneous whitespace. No morphological analyzer or stemmer is employed; this is an intentional design decision to eliminate reliance on external Arabic NLP toolkits.

Phase 4: Class-balanced sample: stratified sampling yields the final corpus of 2855 documents.

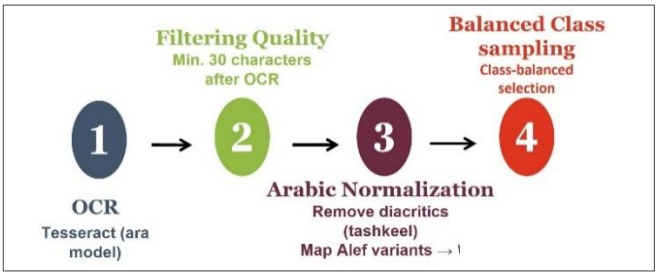


Fig. 1. AraLex-RAG preprocessing pipeline.

TABLE I. Statistics from the AraLex-RAG Corpus

Property	Value
Total documents	2855
Judicial classifications	26
Document nature	Free-form, unstructured Arabic text
Input formats	PDF, DOCX, scanned image (JPEG/PNG)
Mean document length	~420 tokens
Median documents per category	71
Min / Max per category	1 / 602
Categories with < 30 docs	2 out of 26 (Single document)
Train/test split	80% / 20% (stratified)
Cross-validation	5-fold stratified (K=5, seed=42)
Storage backend	SQLite relational database

1) Distribution of classes and imbalance: Table I represents statistical data of the corpus, whereas Table II specifies how it is distributed by class. The dataset exhibits significant imbalance: groups range from a minimum of 1 to a high of 602. Two classifications (1619, 1608) each comprise a single record and are omitted from classifier assessment. The predominant classes, Judicial divorce "التطليق"(602 docs) and النفقة (552 docs), overshadow the corpus and enhance accuracy while diminishing macro F1, as elaborated within the discussion Section.

This disparity is intrinsic to the actual distribution of case categories in Moroccan courts and cannot be rectified via sampling without excluding the already limited minority-class instances. For assessment, an 80/20 stratified division maintains class ratios in both segments; 2853 records that meet quality standards are used for classification and retrieval trials.

TABLE II. ARALEX-RAG SYSTEM DISTRIBUTION OF DOCUMENTS BY CATEGORY

Code	Category	Number
Civil		
1201	Miscellaneous civil المدني المتنوع	86
1202	Tort liability المسؤولية التقصيرية	75
Family		
1601	Civil status amendment الإصلاح والتغيير	161
1602	Birth registration	50
1603	Death registration تسجيل الوفاة	88
1606	Alimony النفقة	552
1607	Judicial divorce التطليق	602
1608	Unilateral divorce الطلاق	1
1609	Child custody الحضانة	56
1610	Spousal return الرجوع لبيت الزوجية	75
1611	Proof of marriage ثبوت الزوجية	70
1612	Right of family access صلة الرحم	65
1613	Lineage / paternity النسب	80
1614	Family enforcement order التذييل بالصيغة التنفيذية (أسرة)	65
1616	Guardianship of minors نواب القاصرين	55
1617	Guardianship of neglected children كفالة الأطفال المهملين	60
1618	Permission for polygamy الإذن بالتعدد	50
1619	Personal status الأحوال الشخصية	1
1620	Other personal status cases قضايا الأحوال الشخصية الأخرى	62
1623	Legal incapacitation التحجير	71
Labour		
1501	Labour disputes نزاعات الشغل	57
1502	Work accidents حوادث الشغل	90
Real Estate		
1401	Standard real estate العقار العادي	80
1402	Registered real estate العقار المحفظ	101
1403	Real estate in registration العقار في طور التحفيظ	102
Rental		
1303	Rent payment and eviction الأداء والإفراغ	100

B. Overview of System Architecture

AraLex-RAG system consists of five operational components: OCR, retrieval, categorization, generation, and interface, supported by a dual-layer storage backend composed of a SQLite relational database that holds all the textual content and associated information of the documents (fields: id, code, label, filename, filepath, file_number, year, decision_number, clean_text, full_text), alongside an FAISS vector index that retains the dimensional dense vector embeddings. Fig. 2 illustrates the whole design and the interconnections among these aspects.

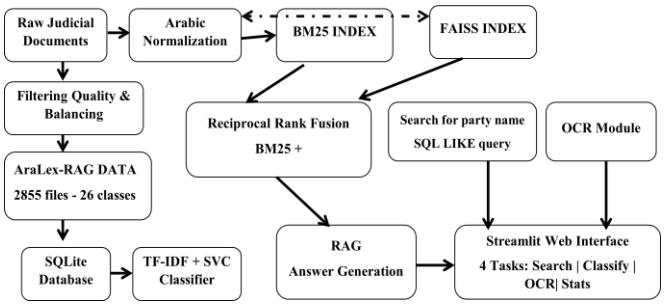


Fig. 2. Overview of AraLex-RAG system architecture.

Corpus indexing constitutes a singular offline procedure. Unstructured files, PDFs, DOCX formats, and scanned pictures are processed using the OCR module (Tesseract utilizing the Arabic language models) and undergo the four-phase normalization process outlined in Section A.2. Every standardized text is thereafter portrayed in two complementary formats: the BM25 engine constructs a token-level inverted index for sparse extraction, whilst the multilingual-e5-small encoder generates a dense embedding preserved in FAISS. Both visualizations come from the same standardized text, guaranteeing that the two retrieval methods function on equivalent input. All textual content, metadata, and embeddings are retained in the storage backend and used for each query.

During inquiry time, the user asks a question using natural language via the interface. The inquiry is subjected to the same normalization as the corpus, thereafter proceeding via a five-stage pipeline shown in Fig. 2. Initially, it is sent concurrently to the BM25 engine and the FAISS index, with BM25 ranking based on word frequency adjusted by inverse document frequency, while FAISS ranking relies on inner product similarity among embeddings. Both ranked classification lists are then integrated by Reciprocal Rank Fusion, which merges the distinct advantages of sparse and dense retrieval without necessitating score adjustment. The categorization module then utilizes the TF-IDF + LinearSVC model to forecast the legal category among the 26 options to which the question applies, hence offering a field labeling that contextualizes the obtained findings. Ultimately, when the generation unit is engaged, the first five extracted files are compiled into a coherent Arabic prompt and presented to LLaMA-3.3-70B via the Groq API, which generates a natural-language response based only on the collected choices. The whole local pipeline, including indexing, data extraction, and categorizing, does not need a GPU; only the RAG phase employs distant inference, allowing the system to remain operable on conventional office equipment.

C. Information Retrieval Algorithms

1) BM25 sparse information retrieval: The BM25 component employs an inverted index based on normalized document tokens. The relevance score is determined by the query Q = {q_1, …, q_m} and the document D.

$$\text{Score}(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)} \quad (1)$$

Q: The query including terms.

D: This is the document that is being evaluated.

$f(q_i, D)$: Term frequency. The frequency of the query phrase q_i in document D.

|D|: Document D length (aggregate word count).

avgdl: Average length of documents throughout the whole corpus.

k_1: A parameter for term frequency saturation. Augmenting this value amplifies the influence of phrase frequency. It's set here to 1.5 ($k_1 = 1.5$). (Standard Okapi BM25 defaults [14]).

b_1: The parameter associated with length normalization. It manages the extent to which the length of a document impacts the score. It's set here to 0.75 ($k_1 = 0.75$). (standard Okapi BM25 defaults [14]).

$IDF(q_i)$: Inverse Document Frequency. Terms are weighted according to how rare they are throughout the corpus. Usually, it is calculated as:

$$IDF(q_i) = ln\left[\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right] \quad (2)$$

In this context, N represents the aggregate count of documents inside the corpus, whereas $n(q_i)$ denotes the quantity of documents that include the phrase q_i.

2) Dense semantic retrieval: FAISS: Dense retrieval utilizes the intfloat/multilingual-e5-small sentence transformer (386 dimensions) [15, 16] to encode both files and queries. The compact version was chosen for resource-efficient implementation: with 2855 documents, the whole FAISS index occupies about 4.3 MB of RAM, allowing sub-second search on conventional hardware without the need for a GPU. Semantic similarity is represented by cosine similarity, which corresponds to the inner product of L2-normalized vectors.

$$sim(u, v) = \frac{u \cdot v}{\|u\| \cdot \|v\|} = \hat{u} \cdot \hat{v} \quad (3)$$

where,

$u \cdot v$: represents the scalar product of the two vectors.

$\|u\|$ and $\|v\|$: the Euclidean norms (magnitudes) of the respective vectors.

Cosine similarity quantifies the angle between two vector representations inside the embedding space: a score of 1 signifies semantically equivalent texts, while a score of 0 denotes not related content. Normalizing each vector corresponding to unit length (L2 normalization) prior to storage guarantees that document length does not enlarge the score, allowing a concise judgment that is entirely relevant to take precedence over an extensive ruling that just briefly references the same words. This normalization also minimizes inner-product searches in FAISS to a cosine similarity query without any additional calculations at the time of querying.

Document vectors are maintained in an FAISS IndexFlatIP [17,19], which does precise inner-product searches with $O(N \cdot d)$ time complexity, with N being the count of indexed documents and d denoting the dimensionality of the embeddings. The index functions on L2-normalized vectors, making inner-product search theoretically synonymous with cosine similarity ranking, therefore necessitating no further transformation during query execution.

3) Reciprocal rank fusion: BM25 and dense retrieval generate relevance ratings on dissimilar scales: BM25 ratings are unrestricted integers weighted by phrase frequency, while cosine similarity scores are confined to the range of [0, 1]. Integrating them via a weighted linear sum necessitates manual scale calibration on reserved data, an unfeasible limitation for a corpus that changes as fresh judgments are included. Reciprocal Rank Fusion (RRF) [10] bypasses this issue by ignoring score magnitudes and combining ranked lists only based on their ordinal positions.

$$RRF(d) = \sum_i \frac{1}{k + rank_i(d)} \quad (4)$$

where, $rank_i(d)$ is the placement of document d in the ordered list generated by retriever i, and k = 60 represents the smoothing constant from the initial formulation [10]. The parameter k controls the rate at which the score diminishes across ranks: with k = 60, a document ranked first contributes a score of 1/61≈ 0.016, whereas a document at rank 10 contributes 1/70 ≈ 0.014, resulting in a carefully moderated decline that ensures the entire top-10 range is treated almost equally, thus preventing the scenario where a single retriever excessively favors a limited selection of documents with extremely high scores.

In AraLex, after every query, two lists are merged: the BM25 ranked list derived from the inverted index and the FAISS cosine similarity ranked list. For any document included in either list, its RRF score aggregates $1/(60 + rank_i)$ from each retriever in which it is found; documents not included in a retriever's list do not contribute from that retriever. Documents that get high rankings from both systems, characterized by lexical and semantic relevance, combine contributions from both lists and hence increase to the highest level of the combined ranking. Fig. 3 shows the whole five-stage query process.

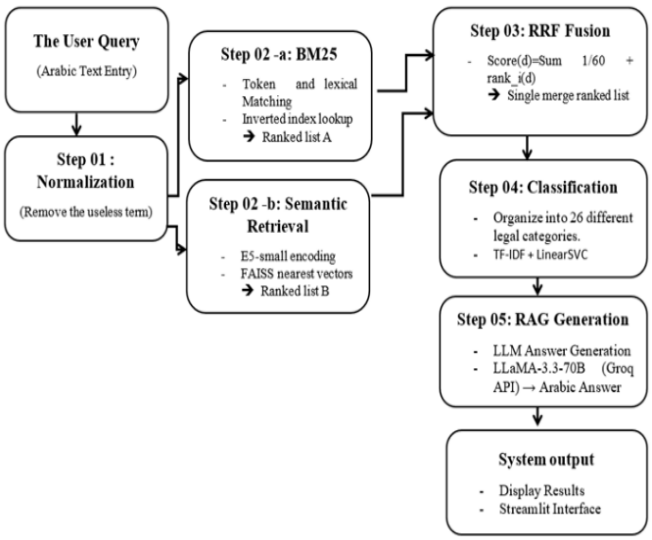


Fig. 3. AraLex-RAG pipeline procedure for query preprocessing.

D. Categorisation Algorithms

Document categorization is executed via a TF-IDF vectorizer in conjunction with a LinearSVC (Linear Support Vector Classifier). The representation of features is constructed using character n-grams instead of words. Arabic has a high degree of morphological complexity: a single root may produce several surface forms by prefixation, suffixation, and cliticization, resulting in word-level characteristics that regard morphological variations as completely distinct. Character n-grams ranging from lengths 2 to 4, meaning ngram_range=(2,4), automatically identify common substrings across these variations, eliminating the need for a stemmer or morphological analysis. The `char_wb` analyzer derives n-grams at word borders, augmenting each term with spaces to ensure that sequences of characters at the beginning and end of words are distinctly represented. Every term t within document d is assigned a TF-IDF [18] weight:

$$w(t,d) = tf(t,d) \times log\left(\frac{N}{df(t)}\right) \quad (5)$$

where, tf(t, d) indicates the occurrence of n-gram t inside document d, N represents the total count of documents in the corpus, and df(t) indicates the quantity of documents that include t. The significance is elevated for n-grams that are prevalent in the present text but infrequent across the corpus, embodying the unique character traits of each legal classification. Each document is thus shown as a sparse vector in a high-dimensional space composed of these weights. LinearSVC was chosen as the classifier due to its exceptional compatibility with high-dimensional sparse feature spaces: it trains significantly faster than kernel-based SVMs while maintaining similar accuracy in text classification tasks. The LinearSVC optimizes the following hinge-loss objective:

$$min_w \, ½\|w\|^2 + C \cdot \sum_i max(0, 1 - y_i(w^\top x_i + b)) \quad (6)$$

where, w represents the weight vector that defines the decision limit, x_i stands for the TF-IDF feature vector for document i, $y_i \in \{-1, +1\}$ signifies its class label, and b is the bias term. The initial component, $½\|w\|^2$, serves as the regularization penalty that maintains a broad and generalizable decision boundary. The second component imposes penalties on documents that exceed the margin, with a weight of C = 1.0, the normal setting, which is effective in high-dimensional feature spaces when the feature count significantly surpasses the document count. Given that SVM is fundamentally binary, the 26-category issue is partitioned into 26 autonomous one-versus-rest (OvR) sub-problems: each classifier is tasked with differentiating one lawful category from the rest. During the prediction phase, all 26 classifiers operate concurrently, and the category with the greatest decision margin is selected for the document.

E. Retrieval-Augmented Generation

The RAG module bases LLaMA-3.3-70B replies on extracted Moroccan judicial rulings, according to the framework established by Lewis et al. [11]. During the inference phase, the process unfolds in three stages. Initially, the user's request is analyzed by the hybrid RRF retrieval system, and the leading k = 5 docs are chosen. Secondly, these documents are compiled into a systematic Arabic prompt using the following standardized format: 1) a directive instructing the model to react only based on the supplied context and to reply in Arabic; 2) the five retrieved documents enumerated and concatenated; 3) the user's query. Third, the compiled prompt is sent to LLaMA-3.3-70B over the Groq cloud API with a temperature setting of $\tau = 0.1$. The reduced temperature setting decreases linguistic variability in outcomes and is suitable for factual legal inquiries where uniformity is prioritized above inventiveness. If the acquired records lack pertinent information on the inquiry, the model is directed to clearly indicate that the context fails to provide an adequate response, which exemplifies appropriate grounded behaviour and inhibits hallucination. The k = 5 context window was selected to strike a compromise between providing enough evidence and maintaining the prompt inside the model's contextual limitations. LLaMA-3.3-70B was chosen over smaller approaches because of its native support for Arabic and its 70 billion parameter scale, which offers enough capability for multi-document legal analysis. The Groq API delivers sub-second inference without the need for local GPU hardware, enabling deployment on conventional office computer systems. LLaMA-3.3-70B is an open-source model, hence avoiding the data privacy issues associated with transmitting sensitive legal documents to private API vendors.

F. Overview of Hyperparameter Settings

Table III summarizes the parameter configurations for each AraLex-RAG component, according to the sequence of the pipeline: sparse retrieval, dense retrieval, rank fusion, classification, and generation. All parameters were either derived from recognized defaults in the referenced literature or chosen according to the limitations of the intended deployment setting.

TABLE III. HYPERPARAMETER CONFIGURATIONS FOR ALL ARALEX-RAG COMPONENTS

Module	Parameter	Setting	Reasoning
BM25	k_1	1.5	Standard Okapi BM25 [14]
	b	0.75	
Dense encoder	Model	multilingual-e5-small	Arabic support and CPU-efficient
	Dim	386	Concise index (<3 MB)
FAISS	Index type	IndexFlatIP	Exact search; N=2855
RRF	k	60	Initial formulation [10]
TF-IDF	analyzer	char_wb	Within-word boundary n-grams
	ngram_range	(2, 4)	Morphological affixes in Arabic
LinearSVC	C	1.0	Standard regularization
	strategy	OvR (26 classes)	OvR multi-class
RAG	Model	LLaMA-3.3-70B	Groq API, temperature=0.1
	Context docs	5	Top-k retrieved

G. Evaluation Framework

The efficacy of classification was assessed using five-fold stratified cross-validation (K=5, random_state=42) on the 2853 documents categorized with a minimum of two samples. The remaining papers were not included since stratified splitting demands at least one sample per class for each fold, a requirement unattainable by single-instance classes. K=5 was chosen as the optimal balance between bias and variance in cross-validation: fewer folds (e.g., K=3) provide bigger test sets but increase variation in the estimation, while more folds (e.g., K=10) are resource-intensive and provide decreasing benefits on a dataset of this magnitude. Stratification guarantees that the class allocation in each fold mirrors the overall corpus distribution, which is essential due to the distinction within groups, with "التطليق" providing 602 documents while multiple categories contribute less than 10. In the absence of stratification, a fold may intentionally lack instances of a rare category, causing the accuracy assessment to be unreliable.

The primary measures are mean accuracy and mean macro F1, precision, and recall over the five folds, shown with a standard deviation. Macro averaging allocates equivalent importance to each category, irrespective of their size, hence imposing the same penalties on the classifier for mistakes in both infrequent and common categories, serving as a more insightful metric than weighted F1 in scenarios of class imbalance. The standard deviation across folds measures the stability of results: a limited range indicates that the classifier generalizes reliably instead of overfitting to a specific data subset.

A distinct fixed 80/20 stratified division (random_state=42) was implemented to derive per-class recall, precision, and F1 metrics throughout all 26 categories. Cross-validation reports only total metrics; the reserved subset offers the detailed analysis necessary to figure out which legal classifications the classifier manages well and which need more data acquisition.

Extraction query and document pairings were generated automatically by the use of template-driven Arabic question formulation using the document metadata. For every document, a collection of Arabic queries was formed based on structured elements such as subject, judgment, year, case identifier, and the name of the court, with each template used only for papers that had the necessary field. This automated method removed a demand for human query labeling and generated 3936 query–document pairings inside the 2855-document corpus. The definitive pertinent document for each inquiry is the original document from which the question was derived, providing one affirmative per query.

The assessment of retrieval was undertaken on these 3936 pairings using Mean Reciprocal Rank [23] (MRR@10) and Normalised Discounted Cumulative Gain [24] (NDCG@10):

$$MRR@10 = \frac{1}{|Q|} \cdot \sum_i \frac{1}{rank_i} \quad (7)$$

$$NDCG@10 = \frac{DCG@10}{IDCG@10}, DCG@10 = \sum_n \frac{rel_n}{log_2(n+1)} \quad (8)$$

where, |Q| = 3,936 is the total number of assessment queries, and $rank_i$ refers to the place of the correct document in the list ranked for query i. A valid document that does not rank within the top 10 adds nothing to the total. MRR@10 consequently favors systems that position the accurate document as near to rank 1 as possible among the top 10 outcomes.

DCG@10 combines pertinence scores marked down by location: rel_n signifies the pertinence of the document at position n (1 if it is the accurate document, 0 if not), while the $log_2(n+1)$ denominator punishes the value of pertinent documents located in lower positions. IDCG@10 is the optimal DCG, the highest attainable score when the accurate document occupies the top position. With one affirmative response per query, IDCG@10 equals $1/log_2(2) = 1.0$; thus NDCG@10 simplifies to $1/log_2(rank_i + 1)$ when the accurate document is located within the top 10, and 0 in all other cases. Dividing by IDCG aligns the score to the range of [0, 1], facilitating comparability across searches.

Furthermore, Hit at 5 (Hit@5) evaluates the presence of the right document within the top 5 findings (1 for yes, 0 for no), averaged over any query. Precision at 1 (P@1) assesses whether the right document is positioned first (1 for yes, 0 for no), averaged across all of the searches. MRR@10 and NDCG@10 assess the quality of rankings, whereas Hit@5 and P@1 evaluate retrieval coverage and precision at the most stringent threshold. RAG generation underwent evaluation using a stratified sample of 45 question-and-answer pairs, chosen to include various sorts of legal queries. The number 45 was established due to the practical limitation of manual ground-truth retrieval, necessitating cautious examination of each source file to get precise reference responses, which-were manually obtained from the relevant source materials, a conventional method in generative QA assessment, whereby human-authored references serve as the norm for comparing system outputs. Generation quality was evaluated using three complementary measures: Token F1 assesses the overlap of words between the generated output and the reference reply; ROUGE-L measures the longest common subsequence [25], rewarding responses that retain the correct order of words; and Cosine Similarity determines semantic similarity through E5 embeddings, ensuring that paraphrases of the correct answer receive high scores even with differing phrasing. All trials were executed on a typical laptop lacking a GPU, indicating that AraLex is operable without dedicated hardware.

IV. RESULTS

A. Classification Outcomes

Table IV displays the outcomes of cross-validation classification. The classifier attains an accuracy of 92.87% ± 1.33% and a macro F1 score of 76.15% ± 3.89%. This difference between accuracy (92.87%) and macro F1 (76.15%) indicates class imbalance: accuracy is influenced by well-supported predominant classes, whereas macro F1 treats all 26 categories roughly the same, resulting in minority classes with fewer test samples adversely affecting the average.

TABLE IV. CLASSIFICATION OUTCOMES: FIVE-FOLD STRATIFIED CROSS-VALIDATION

Metrics	Accuracy	F1-score macro	Precision macro	Recall macro
Mean	92.87%	76.15%	80.27%	75.08%
Std Dev	±1.33%	±3.89%	—	—
Minimum Fold	90.86% (F5)	—	—	—
Maximum Fold	94.20% (F4)	—	—	—

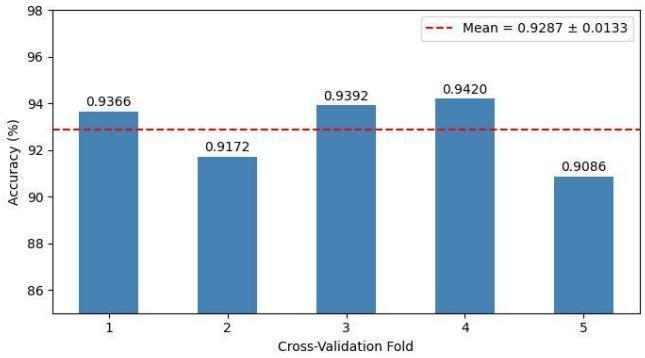


Fig. 4. Fold-specific accuracy throughout five cross-validation folds

Fig. 4 shows a small fold-to-fold range (3.34 percentage points), indicating that the classification algorithm maintains stability and does not exhibit overfitting to any specific data subset. The minimum fold (F5: 90.86%) and the highest (F4: 94.20%) vary by under one standard deviation.

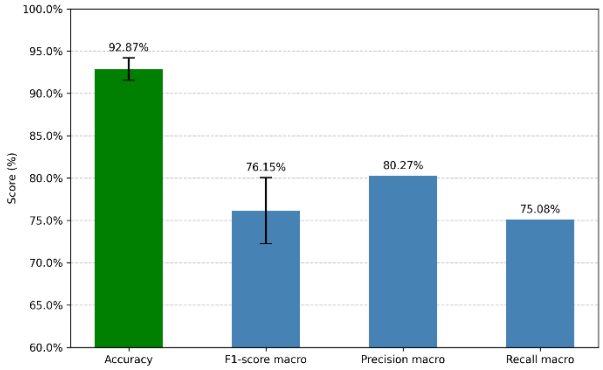


Fig. 5. System classification performance in 26 different legal areas.

Fig. 5 illustrates the four classification measures derived from 5-fold stratified cross-validation. The precision and recall macros are derived from the fixed 80/20 held-out split, whereas the accuracy and F1 macros include standard deviation error bars across folds.

B. Retrieval Outcomes

Table V presents a comparison of BM25-only, semantic-only, and hybrid RRF retrieval over 3936 query-document pairings. Semantic search frequently surpasses BM25 on all four parameters, whereas hybrid RRF attains moderate performance.

The aggregate metric numbers indicate the fundamental challenge of this assessment framework: the 3936 queries use template-driven queries (for instance, " What is the court's ruling?" means "ما هو حكم المحكمة؟") utilized across all sources. Due to the prevalence of analogous language across several legal categories, precise document retrieval using a general query is much more challenging than retrieval via document-specific questions.

As shown in Table V and Fig. 6, the semantic retriever provides the most significant relative enhancement compared to the BM25 baseline, suggesting that semantic representations greatly strengthen retrieval efficacy in the adversarial template-query context. Embracing this enhanced foundation, the hybrid RRF methodology significantly elevates retrieval efficacy, attaining the highest absolute scores across every assessment metric. The supplementary improvements over the semantic retriever are moderate (7–17%), although they remain constant across all assessed measures.

TABLE V. RETRIEVAL EVALUATION

Technique	BM25 only	Semantic only (E5)	Hybrid RRF
P@1	0.002	0.006	0.007
MRR@10	0.005	0.009	0.010
NDCG@10	0.008	0.014	0.015
Hit@5	0.006	0.011	0.012

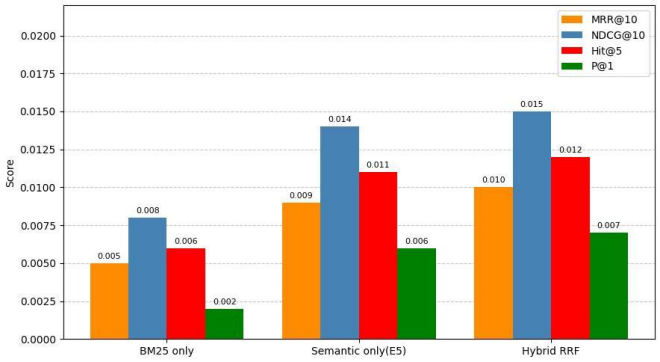


Fig. 6. Retrieval techniques comparison on AraLex corpus.

Category-specific examination reveals significant disparities in retrieval challenges. Minor categories like Permission for polygamy (n=50) attain well, due to the limited pool of available records. Extensive classifications like alimony (n=552) and judicial divorce (n=602) attain lower scores, since several records within these groups use basically the same terms, which complicates the identification of a particular document from a standard template inquiry.

C. Retrieval-Augmented Generation Outcomes

Table VI displays the assessment outcomes of RAG creation for 45 question-and-answer pairs. LLaMA-3.3-70B was accessed using the Groq API for every pair, with the five most relevant files given as context.

TABLE VI. RETRIEVAL-AUGMENTED GENERATION EVALUATION

Measure	Token F1 (SQuAD-style)	ROUGE-L	Cosine Similarity
Parameter	0.261	0.191	0.845
± Std Dev	0.091	0.065	0.062
Analysis	Conceptual overlap vs. fundamental truth	Subsequence with the longest in common	Semantic proximity (E5-embeddings)

Table VI presents the quality of RAG generation evaluated by three complementary measures. Cosine Similarity (0.845) demonstrates a robust semantic correspondence between the produced and reference responses, showing that LLaMA-3.3-70B reliably comprehends the essence of the accurate answer despite variations in phrasing. The values of Token F1 (0.261) and ROUGE-L (0.191) are diminished, which is anticipated for Arabic generative outputs. The model often chooses to paraphrase instead of directly replicating the reference response, adversely impacting word-overlap metrics while maintaining semantic similarity.

D. Benchmarking with Prior Work

Table VII positions AraLex-RAG within the context of comparable legal NLP systems. No direct quantitative comparison can be made among these rows, since each system functions inside a distinct language, domain, class count, and assessment process. The table is designed to demonstrate the range of each work rather than to evaluate performance.

TABLE VII. PLACEMENT OF ARALEX-RAG SYSTEM WITHIN RELATED WORKS

System	Technique	Experiment	Country / Field	Result
Our System (AraLex-RAG)	BM25 + FAISS + RRF	Retrieval + Classification RAG Question-Answering	Morocco 26 classes	Acc=92.87%
	TF-IDF + LinearSVC		Legal decisions	F1=76.15%
	LLaMA-3.3-70B (RAG)		(Table II)	Cosine(RAG)=0.845
[1]	BM25 + BERT re-rank	Legal Case Retrieval (Task1)	Canada Case Law	F1=0.3001 (THUIR)
[5]	TF-IDF + SVM	Prediction Charge	China - CAIL	Acc=94.00%
			criminal cases	
[6]	GloVe + CNN	Judgment prediction	Morocco Six classes	Acc=80.51%
			Court of Cassation	
[7]	TF-IDF + SVM	Legislative Docs classification	Morocco Seven legislative data fields	F1=94.45% Acc=94.58%,
			Document types	
[13]	GPT-4 (RAG)	Arabic question answering (RAG)	Arabic dictionary	Cosine=0.87 F1=0.90
			(structured)	

The systems included in Table VII cover three languages for different nations, illustrating the increasing global attention to legal NLP across various legal frameworks. Each pertains to a distinct objective: case retrieval, judgment forecasting, legislation categorization, or inquiry resolution, under varying resource constraints and assessment methods. AraLex-RAG enhances this domain by focusing on Moroccan Arabic judicial rulings, a context that combines limited resources with the unique language and legal attributes of the Moroccan judicial court.

V. DISCUSSION

TF-IDF combined with LinearSVC achieves 92.87% accuracy without doing any morphological analysis. The justification is clear: Moroccan legal classifications employ remarkably uniform procedural terminology, and character n-grams (n=2,4) inherently detect these subword structures, encapsulating morphological affixes without the need for a stemmer and maintaining resilience when identical roots manifest in various surface forms. Macro F1 (76.15%) is inferior to accuracy since it assigns identical weight to all 26 classes, including those with fewer test samples. The disparity between recall and precision (75.08% vs 80.27%) indicates that the classifier prioritizes caution, overlooking some instances rather than incorrectly labelling them, which is the appropriate compromise for a legal framework.

The retrieval statistics need a contextual framework. The assessment prompts are uniform template inquiries used across all papers, posing the query " What is the court's ruling?" and requiring the system to pinpoint a singular document among hundreds that exhibit the identical procedural language, a process that is fundamentally hard. Under this restricted framework, semantic search exceeds BM25 by 75% on MRR@10, indicating that multilingual-E5-small locates files via semantic relevance rather than mere word correlation, which is crucial for Arabic legal texts because the same idea is expressed in many morphological ways. RAG has a distinct divergence in metrics: Cosine Similarity (0.845) is elevated, indicating that the produced responses are semantically accurate Arabic legal language; however, Token F1 (0.261) and ROUGE-L (0.191) are lower due to the model's tendency to paraphrase instead of replicating the reference response precisely.

The following three limitations should be mentioned properly. The corpus's size suggests that some categories contain fewer than 50 instances, which restricts the classifier and introduces noise into retrieval evaluations; still, with several thousand files per class, targeted document searches

would be practicable, enabling a more earnest exploration of neural techniques. The assessment of retrieval depends on computer-created template questions rather than genuine inquiries from legal professionals; a practitioner study would provide a more realistic depiction of operational performance. In conclusion, the present architecture employs the Groq cloud API for RAG generation, allowing for sub-second inference without reliance on local GPU hardware; a sensible advancement would be to replace it with a locally hosted open-source framework, thus making the system completely autonomous and fit for offline operation in judicial environments. The upcoming phase of action is enhancing the corpus with supplementary decision files that are already in possession.

VI. Conclusion

AraLex-RAG is the initial solution that simultaneously tackles information retrieval, categorization, and RAG question answering for court rulings in Moroccan Arabic. It is constructed upon AraLex-RAG, a set of 2855 unstructured judicial rulings across 26 law domains, and integrates BM25, FAISS dense retrieval, Reciprocal Rank Fusion, character n-gram TF-IDF classification, and LLaMA-3.3-70B creation into a singular program operable on a conventional laptop without a GPU. Assessment validates the feasibility of the methodology: the classifier attains 76.15% macro F1 and 92.87% cross-validation accuracy over 26 legal classifications, hybrid RRF retrieval surpasses both BM25 and semantic search separately, and RAG generation obtains a Cosine Similarity of 0.845, signifying robust semantic coherence among produced and reference responses. These findings indicate that lightweight, GPU-independent techniques may provide robust results on specialized Arabic legal documents, and that AraLex-RAG offers a feasible, implementable option for Moroccan legal practitioners and users without necessitating costly infrastructure.

Acknowledgment

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/25).

References

[1] Kim, M.Y.; Xu, Y.; Goebel, R. (Eds.) Competition on Legal Information Extraction/Entailment (COLIEE) 2023. Lecture Notes in Computer Science; Springer: Berlin, Germany, 2023.

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in Findings of EMNLP 2020, Online, 2020, pp. 2898–2904.

[3] R. Upadhya and T. Y. S. S. Santosh, "LexCLiPR: Cross-lingual paragraph retrieval from legal judgments," in Proc. 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025), Vienna, Austria, 2025, pp. 13971–13993.

[4] S. A. Mahmoudi, C. Condevaux, G. Zambrano, and S. Mussard, "Article 700 identification in judicial judgments: Comparing transformers and machine learning models," Stats, vol. 7, 2024, pp. 1421–1436.

[5] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, et al., "CAIL2018: A large-scale legal dataset for judgment prediction," arXiv preprint arXiv:1807.02478, 2018.

[6] J. Zahir, "Prediction of court decision from Arabic documents using deep learning," Expert Syst., vol. 40, p. e13236, 2023, doi:10.1111/exsy.13236.

[7] A. Bouhouche, M. Esghir, and M. Errachid, "Classification of Moroccan legal and legislative texts using machine learning models," Int. J. Adv. Comput. Sci. Appl., vol. 15, pp. 1108–1114, 2024.

[8] F. Hijazi et al., "ArabLegalEval: A multitask benchmark for assessing Arabic legal knowledge in large language models," in Proc. ArabicNLP 2024, Miami, FL, USA, 2024.

[9] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. T. Yih, "Dense passage retrieval for open-domain question answering," in Proc. EMNLP 2020, Online, 2020, pp. 6769–6781.

[10] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in Proc. SIGIR 2009, Boston, MA, USA, 2009, pp. 758–759.

[11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Adv. Neural Inf. Process. Syst., vol. 33, pp. 9459–9474, 2020.

[12] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, pp. 1–38, 2023, doi:10.1145/3571730.

[13] R. Al-Rasheed et al., "Evaluating RAG pipelines for Arabic lexical information retrieval: A comparative study of embedding and generation models," In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, 2025, pp. 155–164.

[14] S. Robertson and H. Zaragoza," *The probabilistic relevance framework: BM25 and beyond.*". Now Publishers Inc, vol. 4, 2009.

[15] L. Wang, N. Yang, X. Huang, et al., "Text embeddings by weakly-supervised contrastive pre-training," arXiv preprint arXiv:2212.03533, 2022.

[16] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 text embeddings: A technical report," arXiv preprint arXiv:2402.05672, 2024.

[17] Johnson J, Douze M, and Jégou H. "Billion-scale similarity search with GPUs". IEEE transactions on big data. Vol.7, 535-547.

[18] S. Ibrihich, A. Oussous, O. Ibrihich, and M. Esghir, "A Review on Recent Research in Information Retrieval," *Procedia Computer Science*, vol. 201, pp. 777–782, 2022.

[19] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The Faiss Library," *IEEE Trans. Big Data*, 2025, doi:10.1109/TBDATA.2025.3618474. (early access)

[20] N. Y. Habash, Introduction to Arabic Natural Language Processing. San Rafael, CA, USA: Morgan & Claypool, 2010.

[21] Magesh, Surani, Dahl, Suzgun, Manning, Ho, "Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools," Journal of Empirical Legal Studies, 2025, pp. 1–27.

[22] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni, "The Information Retrieval Process," in *Web Information Retrieval*, Berlin, Germany: Springer, 2013, pp. 13–26.

[23] N. Craswell, "Mean Reciprocal Rank," in *Encyclopedia of Database Systems*, L. Liu and M. T. Özsu, Eds. Boston, MA, USA: Springer, 2009, p. 1703.

[24] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," ACM Trans. Inf. Syst., vol. 20, no. 4, pp. 422–446, 2002.

[25] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Proc. ACL-04 Workshop: Text Summarization Branches Out, Barcelona, Spain, 2004, pp. 74–81.

IoT-Enabled Smart Cap for Real-Time Stress Monitoring

Bairavanathan Maha Sri[1], Kala Devi Managuran[2], Rajermani Thinakaran[3], Narayana Lokesh T[4], Punitha Shri M[5], Jeya Mithra K[6], Meenakshi B[7], Raja Subramanian R[8], Tong Mingjie[9]
Faculty of Data Science and Information Technology, INTI International University, Malaysia, Nilai, Malaysia[1, 2, 3, 9]
Kalasalingam Academy of Research and Education, Tamil Nadu, India [4, 5, 6, 7, 8]

***Abstract*—Stress is a physiological and psychological response that can affect the well-being and daily functioning of an individual. In this study, we propose an IoT-enabled smart cap for real-time stress monitoring using multiple physiological signals. The system uses electroencephalography (EEG), electrocardiography (ECG), galvanic skin response (GSR), and pulse measurements to monitor physiological changes associated with varying levels of stress. The sensor data is captured and communicated to the Blynk Cloud platform using a Raspberry Pi 3 Model B. The acquired data is processed by a Node.js backend and also allows real-time display using a web-based dashboard. The physiological measurements are combined with a rule-based stress classification method. Stress is classified as mild, moderate, and severe. Depending on the detected stress level, the system provides appropriate relaxation and intervention suggestions through the monitoring interface. The proposed architecture demonstrates the feasibility of integrating multimodal physiological sensing with IoT-based monitoring in a wearable form factor. The system is intended to provide continuous, accessible, and real-time stress monitoring while supporting personalized intervention. Further validation using larger datasets, diverse participants, and data-driven machine learning or deep learning models is required to establish the accuracy, generalizability, and clinical applicability of the proposed system.**

Keywords*—Stress monitoring; EEG; ECG; GSR; IoT; Blynk cloud platform; mental health management; real-time monitoring; personalized recommendations

I. Introduction

Stress has become an important concern because it can influence both psychological well-being and physical functioning. When stress persists for a long period, it may interfere with daily activities, concentration, and overall health. Stress is commonly assessed through questionnaires, psychological screening, and clinical evaluations [13], [14]. These methods can help determine how an individual experiences stress, but they are generally performed at particular points in time. As a result, changes in physiological responses between assessments may not be captured. Wearable monitoring provides an alternative approach by allowing physiological changes to be observed continuously and by making real-time measurements possible [14], [15].

Conventional approaches to stress assessment and management also have practical limitations when information about changing physiological conditions is required. Psychological assessments depend partly on an individual's responses, while clinical evaluations may not always be immediately accessible. These limitations create a need for systems that can observe physiological changes and present the information without requiring repeated assessments [14]. IoT technology can support such systems by connecting sensors, computing hardware, cloud services, and monitoring interfaces. Through this type of arrangement, physiological measurements can be collected and transmitted for observation while the individual is being monitored [8].

Wearable physiological monitoring has received considerable attention in stress-related applications [14], [15]. Different sensors can capture different aspects of the body's response. EEG records electrical activity associated with brain function, whereas ECG records the electrical activity of the heart. GSR measures changes in skin conductance, and pulse sensing provides information related to cardiovascular activity. Using these measurements together can provide information from several physiological sources instead of relying on one signal alone [17]. For a practical wearable application, however, the collected signals also need to be handled efficiently and made available through a monitoring interface.

Wearable systems developed for stress monitoring still present several challenges. Some approaches concentrate on one physiological signal, while others depend on offline processing or separate devices for different measurements. Another consideration is that physiological responses are not identical across individuals. A measurement that represents a particular state for one person may not necessarily indicate the same state for another person. Factors such as physical activity, surroundings, and an individual's normal physiological condition can also affect the recorded signals [15], [19]. Therefore, classification based on generalized thresholds may have limited reliability when applied across different users [12], [15], [19].

The combination of wearable sensing and IoT connectivity offers a practical way of addressing these requirements. Sensor readings can be obtained continuously, handled by an embedded computing unit, transferred through a network, and made available through a cloud-based interface. Such an arrangement allows physiological information to be viewed while the system is operating rather than only after data collection has been completed. In this study, these components are brought together in a wearable smart-cap architecture using EEG, ECG, GSR, and pulse sensors. A Raspberry Pi is used for data acquisition and transmission, while the Blynk Cloud Platform is used for data visualization and monitoring.

The collected measurements also need to be interpreted so that the user can understand the current stress state. For this purpose, the proposed system uses a threshold-based approach to divide the detected stress state into three categories: mild, moderate, and severe. A corresponding suggestion is displayed through the user interface according to the classified level. The suggestions are intended to provide simple supportive information that may help the user respond to the detected state; they are not intended to constitute medical diagnosis or treatment. The system therefore serves as a monitoring and feedback platform, while medical or psychological assessment remains outside its scope.

The present work focuses on the development of an IoT-enabled smart cap that brings multimodal physiological sensing and real-time monitoring into a single wearable setup. The system connects EEG, ECG, GSR, and pulse sensors to a Raspberry Pi, which handles the acquired measurements and their transmission to the Blynk Cloud Platform. The cloud-connected interface allows the collected information and classification output to be observed during system operation. Rather than using an AI or deep-learning model for the current classification process, the prototype applies predefined threshold-based rules. More advanced machine learning approaches may be investigated in future work to improve the adaptability of the classification process [16], [19].

The main objective of this study is to examine the feasibility of using a wearable IoT architecture for continuous physiological monitoring and three-level stress classification. The work brings together multiple physiological sensors, Raspberry Pi-based data handling, cloud connectivity, real-time visualization, and threshold-based classification within a smart-cap prototype. System operation is examined through the acquisition and transmission of sensor measurements, visualization of the collected data, stress-level classification, and user feedback. The prototype provides a basis for further investigation, particularly through larger-scale data collection, more systematic validation, and the use of learning-based classification methods in future studies.

II. Literature Review

A. Stress Monitoring and Physiological Assessment

Stress has been examined from both psychological and physiological viewpoints because it can influence how people feel, behave, and function physically. Previous studies have looked at stress in different settings, including academic and occupational environments. Sattar et al. [1] studied the relationship between medical professionalism, coping, and the mental well-being of medical students, showing how demanding environments can influence psychological well-being. McDaid et al. [2] examined the economic and social effects of mental disorders in South Asia and discussed the broader consequences associated with limited mental-health support. Together, these studies show why stress and mental well-being require attention. However, their main focus is on psychological, social, and population-level aspects rather than continuous monitoring of physiological changes.

In most cases, stress is assessed through self-reported questionnaires, psychological assessments, clinical evaluations, or a combination of these methods [13], [14]. These approaches can provide useful information about a person's condition, but the results generally describe the person's state at the time of assessment. Physiological monitoring can provide additional information by tracking changes as they occur over time. This has contributed to the use of wearable sensors for monitoring physiological signals outside conventional clinical environments [14], [15].

Wearable and pervasive-computing technologies have also been explored in healthcare because they can collect physiological information continuously. Dewi et al. [3] discussed the use of pervasive computing and wearable devices in sustainable healthcare applications and highlighted the role of connected wearable technologies in health monitoring. IoT-based healthcare systems have also been studied for collecting and transferring health-related information through connected devices [8]. However, a general healthcare monitoring system does not directly address all the requirements of stress monitoring. Stress-related applications also need to consider the selection of physiological signals, the combination of multiple signals, stress classification, and the delivery of timely feedback.

B. Wearable Physiological Sensing for Stress Monitoring

Wearable devices can measure physiological changes that occur when a person responds to psychological or environmental stress. Researchers have explored several types of signals for this purpose, including cardiac activity, electrodermal activity, brain activity, skin temperature, respiration, and physical activity. Bolpagni et al. [5] reviewed the use of bio signals from wearable devices for personalized stress detection and reported the use of different physiological measurements across stress-monitoring studies. Similarly, Vos et al. [12] examined machine-learning approaches for stress monitoring using wearable devices and discussed the importance of physiological data quality and generalizability. These studies indicate that the signals selected for stress monitoring depend on the type of wearable device, the conditions under which the data are collected, and the purpose of the monitoring system.

Cardiac measurements are commonly considered in stress-monitoring studies because changes in cardiac activity can occur during physiological arousal. ECG can be used to record the electrical activity of the heart and can also provide measures such as heart rate and heart-rate variability. Pulse or photoplethysmography-based measurements can provide related information about cardiovascular activity. Electrodermal activity, also known as galvanic skin response (GSR), measures changes in skin conductance and is another commonly used physiological signal in stress-related studies. The review by Bolpagni et al. [5] identified EDA and PPG among the commonly used signals in wearable stress-detection research, while Sosa et al. [17] examined multimodal approaches that combine different physiological indicators of sympathetic nervous system arousal. These findings support the use of multiple physiological signals when monitoring stress rather than relying on a single measurement.

EEG provides another sensing modality because it records electrical activity associated with brain function. It has been investigated for mental-stress assessment because changes in

brain activity may be related to cognitive and emotional states. Badr et al. [4] reviewed deep-learning methods for mental-stress assessment using EEG and discussed approaches based on convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and hybrid architectures. The use of EEG alongside other physiological measurements therefore provides another possible source of information for multimodal stress-monitoring systems [17].

Wearable sensing also brings some practical challenges. Sensor placement, movement artefacts, signal quality, environmental conditions, individual physiological differences, and user comfort can all affect the collected data. Bolpagni et al. [5] highlighted issues including signal quality, dataset representativeness, privacy, and real-world deployment in wearable stress detection. Abd Al-Alim et al. [15] also investigated wearable-sensor-based stress detection in free-living environments, where physiological measurements can be influenced by everyday conditions. In addition, Vos et al. [12] discussed the challenge of developing stress-monitoring models that generalize across wearable datasets and users. These findings suggest that the number of sensors alone does not determine the quality of stress classification; the quality of the collected signals and the way they are interpreted are also important.

Table I summarizes the physiological signals relevant to stress monitoring, the information they provide, and the main considerations associated with their use [5], [17].

TABLE I. PHYSIOLOGICAL SIGNALS RELEVANT TO STRESS MONITORING

Physiological Signal	Information Obtained	Relevance to Stress Monitoring	Considerations
EEG	Electrical activity of the brain	Provides information related to changes in brain activity and mental state	Can be affected by movement and electrical noise
ECG	Electrical activity of the heart	Provides cardiac information and can support heart-rate and HRV analysis	Electrode placement and movement can affect signal quality
GSR/EDA	Changes in skin conductance	Provides information associated with autonomic arousal	Can be influenced by temperature, movement, and skin condition
Pulse	Cardiovascular pulse activity	Provides information related to cardiovascular response	Can be affected by physical activity
Skin Temperature	Peripheral temperature changes	Can provide additional physiological information	Influenced by surrounding environmental conditions
Respiration	Breathing rate and pattern	May change during physiological arousal	Requires appropriate sensor placement

The smart cap uses EEG, ECG, GSR, and pulse measurements so that information can be obtained from several physiological sources. This multimodal configuration is consistent with the broader direction of wearable stress-monitoring research, which has increasingly considered combinations of physiological signals and personalized monitoring approaches [5], [17]. The cap-based design therefore provides a wearable form factor for bringing several sensing modalities together within an IoT-enabled monitoring system.

C. Wearable Devices and Multimodal Stress Monitoring

Wearable devices have developed from systems that measure individual physiological signals into platforms capable of collecting several types of physiological measurements. The Empatica E4, for example, is a research-oriented wearable platform designed to collect multiple biosignals and has been used in studies involving wearable physiological monitoring [11]. Such systems illustrate how physiological measurements can be collected while the user is wearing the device rather than through a single assessment session. In stress-monitoring research, Bolpagni et al. [5] reviewed the use of biosignals from wearable devices and reported that different physiological measurements are used depending on the sensing configuration and monitoring objective. The review also discussed the increasing interest in personalized and multimodal stress-detection approaches.

The placement of a wearable device depends largely on the physiological signal being measured. Different wearable arrangements are used to accommodate the sensing requirements of cardiac, electrodermal, and brain-related measurements. EEG requires electrodes to be positioned on the scalp, making a head-worn arrangement suitable when brain activity is included. These differences in sensing requirements are important when designing a wearable system that combines several physiological signals. Bolpagni et al. [5] discussed practical factors such as signal quality, sensor configuration, and real-world deployment in wearable stress-detection systems. These considerations become particularly relevant when several sensors have to operate together in a single wearable form factor.

Riedl et al. [6] examined EEG and ECG responses in relation to videoconference fatigue. Rountree-Harrison et al. [18] reviewed EEG and ECG biomarkers in traumatic stress research and examined studies using both machine-learning and non-machine-learning approaches. Although fatigue and psychological stress are not the same condition, their study demonstrates the use of more than one physiological signal to examine changes associated with human cognitive and physiological states. Wang et al. [7] investigated wearable sensing technologies for monitoring construction workers, demonstrating the application of wearable sensing in an occupational environment. The studies show that the design and placement of wearable sensors can vary according to the application and the physiological information being collected.

The use of multiple physiological signals has received increasing attention in stress-detection research. Bolpagni et al. [5] identified EDA and PPG among the commonly used biosignals in wearable stress-monitoring studies and discussed multimodal approaches that combine different physiological measurements. Sosa et al. [17] specifically reviewed multimodal approaches for sympathetic nervous system arousal classification and examined how different physiological indicators can be considered together for stress-related

classification. This line of research is relevant to systems that do not rely on a single physiological measurement, since different signals can provide complementary information about physiological responses.

Another important issue is whether a wearable stress-monitoring system can maintain reliable performance across different users and real-world conditions. Vos et al. [12] reviewed machine-learning approaches for stress monitoring using wearable devices and discussed challenges related to generalizability. Abd Al-Alim et al. [15] investigated wearable-sensor-based stress detection in free-living environments, where physiological measurements are collected under less controlled conditions than laboratory settings. These studies suggest that signal collection and system design need to be considered together with the conditions in which the wearable system is used.

Table II compares representative wearable stress-monitoring approaches based on their typical physiological signals, main strengths, and limitations.

TABLE II. COMPARISON OF WEARABLE STRESS-MONITORING APPROACHES

Approach	Typical Signals	Main Strength	Main Limitation
Wrist-based wearable [5]	Pulse/PPG, EDA, motion	Portable and convenient for continuous monitoring	Does not normally provide EEG-based brain activity
Chest-based wearable [5]	ECG, respiration, motion	Suitable for cardiac and respiratory measurements	May be less convenient for everyday use
EEG-based wearable [4]	EEG	Provides information about brain electrical activity for mental-stress assessment	Sensitive to electrode placement and movement
Multimodal wearable [5], [17]	ECG/PPG, EDA, EEG and/or other signals	Combines information from different physiological sources	Requires additional hardware and signal processing
Proposed smart cap	EEG, ECG, GSR, pulse	Combines multiple physiological measurements with IoT monitoring	Current prototype uses threshold-based classification and requires wider validation

The proposed smart cap uses EEG, ECG, GSR, and pulse measurements within a single head-worn arrangement. The inclusion of EEG provides access to brain-related physiological activity, while ECG, GSR, and pulse provide additional physiological measurements. Unlike the commercial and conventional wearable examples discussed above, the proposed system combines these sensing modalities in a cap-based prototype and connects the acquired measurements to an IoT monitoring platform. The purpose is to examine the feasibility of integrating multimodal physiological sensing, IoT connectivity, and real-time monitoring within a single wearable setup rather than to claim superiority over existing wearable devices.

D. IoT Integration and Cloud-Based Monitoring

The Internet of Things (IoT) allows sensing devices to exchange information with computing systems and cloud platforms. In healthcare applications, this can involve several connected components, including sensors, embedded processors, communication networks, cloud services, and user interfaces. Pradhan et al. [8] discussed the use of IoT technologies in healthcare and their role in collecting and transferring health-related information.

For a wearable stress-monitoring system, IoT connectivity makes it possible to move beyond simply collecting data on the device. Sensor readings can first be received by an embedded computing unit and then sent over a network to a cloud-based platform. The information can be displayed through a dashboard, allowing the measurements to be observed while the system is operating.

The Blynk platform used in this study provides the cloud-connected environment for displaying and monitoring the collected measurements [9]. The platform itself, however, does not determine whether a particular physiological response represents stress. The sensor measurements still need to be interpreted using an appropriate classification or decision-making method.

In the developed prototype, EEG, ECG, GSR, and pulse sensors collect the physiological measurements. A Raspberry Pi receives the sensor data and handles its processing and transmission. The processed information is then sent through the IoT connection to the Blynk Cloud Platform, where it can be viewed through the monitoring interface. This allows both the physiological readings and the resulting stress level to be observed while the system is running.

Table III summarizes the functional components of the proposed IoT architecture and describes the role of each layer in the collection, processing, transmission, visualization, and feedback of physiological data.

TABLE III. FUNCTIONAL COMPONENTS OF THE IOT ARCHITECTURE

Layer	Component	Function
Sensing Layer	EEG, ECG, GSR, Pulse	Collects physiological measurements
Processing Layer	Raspberry Pi	Receives and processes sensor data
Communication Layer	IoT/Wi-Fi connectivity	Transfers measurements to the cloud platform
Cloud Layer	Blynk Cloud Platform	Supports data transmission and visualization
Application Layer	Monitoring interface	Displays physiological information and stress level
Feedback Layer	Supportive content	Provides suggestions corresponding to the detected stress level

The architecture brings sensing, processing, communication, visualization, and feedback together within the same monitoring framework rather than treating them as separate components.

E. Machine Learning and Deep Learning for Stress

Stress classification can be approached using predefined rules as well as conventional machine-learning and deep-learning methods. Castro-García et al. [10] reviewed human stress and activity recognition using low-cost wearable technologies and discussed the use of physiological measurements for automated recognition. Their work illustrates how wearable sensing can be combined with computational methods for recognizing stress-related states. More recent research has examined machine-learning pipelines in greater detail. Vos et al. [12] reviewed wearable stress-monitoring studies using machine learning and considered issues such as data labelling, biomarker validity, and the ability of models to generalize to previously unseen data.

The development of a learning-based stress classifier depends not only on the choice of algorithm but also on the data used for training and evaluation. Vos et al. [12] emphasized that generalization and reproducibility remain important challenges in wearable stress-monitoring research. In practical use, physiological signals can differ across individuals and can also be affected by factors such as physical activity and the conditions in which the measurements are collected. Abd Al-Alim et al. [15] addressed this issue by investigating stress detection using wearable sensors in free-living environments rather than relying only on controlled laboratory conditions. Their study demonstrates the importance of evaluating stress-detection approaches under conditions that more closely resemble everyday use.

Deep-learning methods have received particular attention for EEG-based stress assessment. Badr et al. [4] reviewed deep-learning approaches for mental-stress assessment using EEG and discussed CNN, LSTM, and hybrid architectures. The review also examined how different forms of EEG representation are used as inputs to these models. These approaches show the potential of learning complex patterns from physiological signals, but their effectiveness depends on suitable datasets, reliable stress labels, and appropriate evaluation.

The current smart-cap prototype does not use a machine-learning or deep-learning model as its main classification method. Instead, it uses predefined threshold-based rules to divide the monitored state into three levels: mild, moderate, and severe. This approach allows the prototype to demonstrate the complete sensing, processing, IoT connectivity, and real-time classification pipeline without requiring a large labelled dataset for model training. However, fixed thresholds may not account for differences between users. The literature on wearable stress monitoring therefore provides a basis for considering machine-learning and deep-learning methods as future extensions once a sufficiently large and reliably labelled dataset is available.

F. Critical Analysis and Research Gap

The studies reviewed above show a gradual shift from conventional stress assessment toward continuous physiological monitoring using wearable technologies. Bolpagni et al. [5] examined the use of biosignals from wearable devices for stress detection and discussed the growing interest in multimodal and personalized approaches. IoT-based healthcare research has also demonstrated how connected devices can support the collection and transfer of physiological information between sensing units and monitoring platforms [8].

The use of multiple physiological signals can provide information from different aspects of the body's response to stress. Sosa et al. [17] reviewed multimodal approaches for stress-related physiological arousal and examined the use of different physiological indicators together. The availability of wearable platforms such as the Empatica E4 also demonstrates the practical use of multi-sensor physiological monitoring in wearable form factors [11].

Machine-learning and deep-learning methods provide further possibilities for identifying patterns in physiological data. Badr et al. [4] reviewed deep-learning approaches for EEG-based mental-stress assessment, including CNN, LSTM, and hybrid models. However, Vos et al. [12] highlighted generalization as an important challenge in wearable stress-monitoring research. Abd Al-Alim et al. [15] also investigated stress detection in free-living environments, where physiological measurements are collected under less controlled conditions.

These findings indicate that reliable stress monitoring involves more than selecting sensors or classification algorithms. Individual physiological differences, signal quality, environmental conditions, and the availability of suitable labelled data can all affect system performance. A model developed from a limited participant group may therefore not necessarily produce the same results when applied to new users [12].

Another challenge is integrating the different functions required in a practical wearable stress-monitoring system. The system needs to acquire physiological signals, process the measurements, transfer the information, present the results, and provide feedback. The proposed smart cap addresses this integration by combining EEG, ECG, GSR, and pulse sensing with a Raspberry Pi and the Blynk Cloud Platform.

The current implementation uses a threshold-based mechanism to classify the monitored state into mild, moderate, and severe levels. This allows the prototype to demonstrate multimodal sensing, IoT connectivity, real-time visualization, and stress-level classification without requiring a large labelled dataset for model training. However, fixed thresholds may not account for differences between users.

The main research gap addressed by this work is therefore the integration of multiple physiological sensing modalities, IoT-based monitoring, and real-time stress-level feedback within a single cap-based wearable prototype. The system is not intended to claim better performance than existing machine-learning or deep-learning methods. Instead, it provides a prototype that can be further investigated through larger datasets, systematic stress labelling, personalized calibration, and learning-based classification.

III. Methodology

The smart cap was developed by combining physiological sensors with a Raspberry Pi, IoT connectivity, and a monitoring dashboard. The work was carried out in four phases. First, the required physiological signals were collected through the

sensors. The second phase focused on arranging the hardware and software components and defining their data flow. The third phase involved implementing the prototype, followed by functional and usability testing in the fourth phase.

A. Phase 1 - Data Collection and Sensor Integration

Fig. 1 shows the proposed arrangement of the smart cap. EEG, ECG, GSR, and pulse sensors were selected to obtain different types of physiological measurements. The sensors were fitted into the cap-based setup and connected to the Raspberry Pi for data acquisition.

The measurements obtained from the sensors were monitored through the IoT connection. Based on the predefined conditions used in the prototype, the monitored state was grouped into three levels: Level 1 (mild), Level 2 (moderate), and Level 3 (severe).

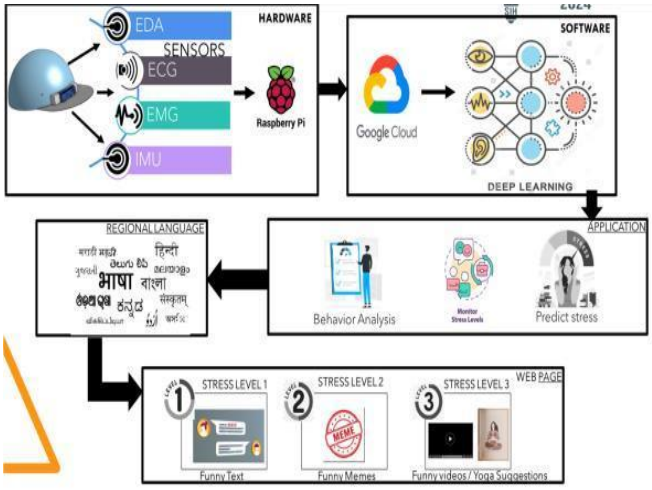


Fig. 1. Full level design of the smart cap.

B. Phase 2 – System Design

Fig. 2 illustrates the top, side, bottom, and isometric views of the proposed IoT-enabled smart cap, providing an overview of its physical structure and component arrangement. The design was divided into four parts: hardware, cloud connectivity, backend processing, and the dashboard.

1) Hardware: The EEG, ECG, GSR, and pulse sensors were connected to a Raspberry Pi 3 Model B. It receives the measurements generated by the sensors and prepares them for further processing and transmission.

2) Cloud connectivity: The sensor information from the Raspberry Pi is sent wirelessly to the Blynk Cloud Platform. Blynk is used in the prototype to monitor and display the incoming measurements.

3) Backend: The Node.js backend receives the relevant sensor information and applies the threshold conditions defined for the prototype. The output of this process is the corresponding stress level.

4) Frontend: The dashboard was developed using HTML, CSS, and JavaScript. It displays the physiological readings together with the detected stress level. The dashboard also provides supportive material, such as relaxation videos, breathing exercises, and yoga suggestions, depending on the displayed stress level.

The main data path used in the prototype is: Sensor → Raspberry Pi → Blynk Cloud → Node.js → Dashboard. Keeping these components separate also makes it easier to modify individual parts of the prototype during further development.

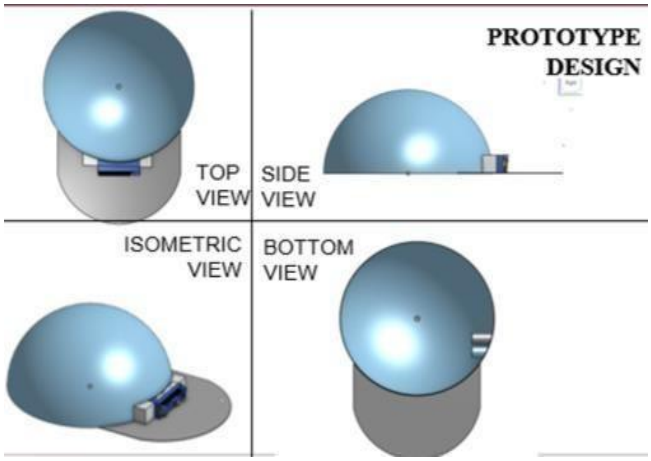


Fig. 2. Top, side, bottom, and isometric views of the proposed IoT-enabled smart cap.

C. Phase 3 – Implementation

During implementation, the sensor connections and software components described in the system design were integrated into the working prototype. The EEG, ECG, GSR, and pulse sensors were connected to the Raspberry Pi and checked for proper data acquisition and communication. The Raspberry Pi program was configured to receive the sensor readings and forward the collected information to the Blynk Cloud Platform at regular intervals. The Node.js backend was used to apply the predefined threshold-based rules and generate the corresponding stress-level output. The dashboard then displayed the sensor information and stress classification.

D. Threshold-Based Stress Classification

The stress level was determined using predefined threshold-based rules applied to the physiological measurements. The classification process can be represented conceptually as

$$S=f(HR,ECG,GSR,P)$$

where, HR, ECG, GSR, and P represent the physiological measurements obtained from the sensors, and Srepresents the resulting stress level. Based on the predefined conditions implemented in the Node.js backend, the output was grouped into three categories: Level 1 (mild), Level 2 (moderate), and Level 3 (severe).

Depending on the detected level, the interface presented the relevant supportive content, including relaxation material, breathing exercises, and yoga-related suggestions. The complete processing path was therefore:

Sensor Signal → Raspberry Pi → Blynk Cloud → Node.js Backend → Frontend Display

E. Phase 4 – Testing

The prototype was tested to check whether the major components operated as intended. The testing covered sensor data acquisition, system response, and interaction with the dashboard.

1) Test case 1: sensor data verification: The sensor readings were observed under the test conditions to check whether measurements could be acquired and transmitted without interruption.

2) Test case 2: system response: Different input conditions were provided to the threshold-based classification component. The resulting stress level was checked against the expected output, and the dashboard response was observed for each case.

3) Test case 3: user interface evaluation: The dashboard was checked for basic usability aspects such as readability, ease of navigation, and response to changes in the displayed stress level. Feedback was also collected regarding the presentation of the physiological information and the supportive content.

Fig. 3 presents the developed prototype of the smart cap, showing the integration and placement of the sensing and processing components within the wearable structure.



Fig. 3. Prototype of the cap.

IV. Results and Discussion

A. Prototype Outcome and Functionality

The developed Stress Monitoring Cap combines EEG, ECG, GSR, and pulse sensors in a cap-based wearable setup. The sensors are connected to a Raspberry Pi 3 Model B, which acts as the main unit for receiving the physiological readings and passing the required information to the connected IoT platform. Fig. 3 shows the developed smart-cap prototype with the integrated EEG, ECG, GSR, and pulse sensing components.

During operation, the sensor readings can be viewed through the monitoring dashboard. The system also produces a stress-level output using the threshold conditions implemented in the Node.js backend. Based on these conditions, the monitored state is displayed as Level 1 (mild), Level 2 (moderate), or Level 3 (severe).

The prototype brings sensor acquisition, data transmission, stress-level classification, and user feedback together in one system. The stress level shown by the prototype represents the result of the implemented classification rules and is not intended to be used as a clinical diagnosis.

B. Software Design and System Performance

Fig. 4 shows the overall architecture of the Stress Monitoring Cap. The system is arranged into different functional stages, beginning with physiological signal acquisition and ending with the presentation of the results through the dashboard.

The EEG, ECG, GSR, and pulse sensors are the starting point of the data flow. Their readings are picked up by the Raspberry Pi, which prepares the acquired information for transmission. The readings are subsequently sent over the wireless connection to the Blynk Cloud Platform. This allows the physiological information to be monitored while the system is operating.

The Node.js backend receives the relevant sensor information and applies the threshold conditions defined for the prototype. The resulting stress category is then passed to the frontend. The dashboard, developed using HTML, CSS, and JavaScript, displays the physiological readings and the corresponding stress level.

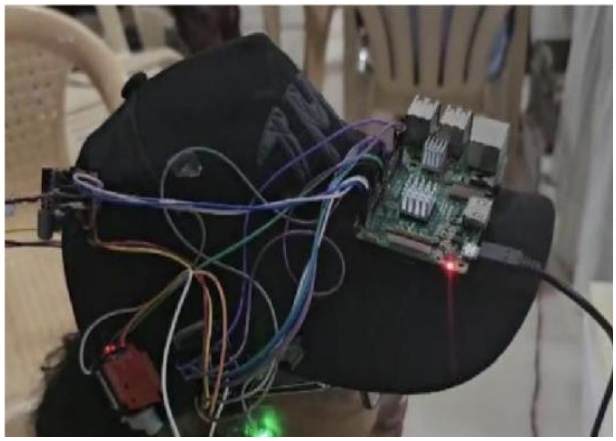


Fig. 4. Overview of the cap.

The interface also provides supportive content based on the detected level. This includes short textual suggestions, relaxation videos, breathing exercises, and yoga-related recommendations. The intention is to provide the user with simple actions that may help them respond to the displayed stress level.

C. Prototype Testing and Observations

The prototype was tested to check the operation of the sensor connections, data transmission, stress-level classification, and dashboard response. The testing focused mainly on whether the individual components could work together through the complete data path.

For sensor-data verification, the EEG, ECG, GSR, and pulse readings were observed during system operation. The Raspberry Pi was checked for its ability to receive the sensor outputs and forward the collected information to the connected platform.

The threshold-based classification was also tested with different input conditions. The resulting output was compared with the expected category defined by the implemented rules.

The dashboard was then checked to ensure that the corresponding stress level and supportive content were displayed.

The dashboard was checked during system operation to see whether the displayed information remained readable and whether the interface responded when the stress-level output changed. Navigation between the available dashboard elements was also checked. These observations were used to confirm that the interface could be used for viewing the sensor readings and the corresponding system output.

D. Discussion

The prototype demonstrates the feasibility of combining multiple physiological sensors with IoT connectivity in a wearable cap. The system receives the sensor measurements and forwards the required information through the wireless connection. Blynk provides the cloud-based monitoring component, while the Node.js backend handles the threshold-based stress classification. Fig. 5 shows the smart-cap monitoring dashboard, where the physiological readings, stress-level output, and supportive content are presented to the user. The web dashboard then presents the measurements and classification result to the user.

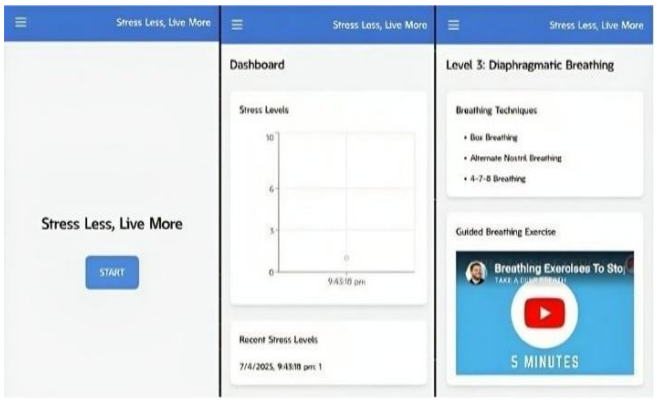


Fig. 5. Smart cap monitoring dashboard.

One practical advantage of the threshold-based approach is that it can be implemented without requiring a large labelled dataset for model training. This is useful for an initial prototype because the complete sensing, communication, classification, and feedback pipeline can be demonstrated without depending on a trained machine-learning model.

However, fixed thresholds also introduce limitations. Physiological measurements can vary from one person to another, and a value that represents an increased physiological response for one user may not have the same meaning for another user. Sensor placement, movement, environmental conditions, and signal noise can also affect the measurements.

Therefore, the current prototype should be considered a feasibility demonstration rather than a clinically validated stress-detection system. The testing carried out in this work mainly examines whether the hardware, communication, classification, and dashboard components operate together as intended. It does not provide sufficient evidence to claim clinical accuracy or generalizable performance across a wider population.

Another limitation is the current use of predefined rules instead of a data-driven classification model. Machine-learning and deep-learning methods could be explored in future versions of the system. With a sufficiently large dataset containing reliable stress labels, models could be trained using the combined physiological signals and evaluated using appropriate training and testing procedures.

The IoT component also provides scope for further development. Continuous cloud-based monitoring could allow the system to maintain longer-term records of physiological measurements and stress-level changes. Such information could be useful for studying individual patterns and developing personalized thresholds or learning-based models.

Overall, the developed prototype demonstrates the integration of multimodal physiological sensing, Raspberry Pi-based data handling, IoT communication, real-time visualization, threshold-based classification, and supportive feedback within a wearable cap. Further validation with a larger and more diverse participant group would be required before assessing its effectiveness for real-world or clinical applications.

V. Conclusion and Future Enhancement

The Stress Monitoring Cap was developed as a wearable prototype for monitoring physiological signals related to stress. EEG, ECG, GSR, and pulse sensors were integrated with a Raspberry Pi 3 Model B for data collection and processing. The measurements were transmitted through the Blynk Cloud Platform and presented on a web-based dashboard. Based on the predefined threshold conditions, the system displays the detected state as mild, moderate, or severe. The dashboard also provides relaxation videos, breathing exercises, and yoga-related suggestions according to the displayed level.

The prototype shows that multiple physiological signals and IoT connectivity can be combined in a single cap-based monitoring system. The present implementation is mainly intended to demonstrate the working of the proposed system. It does not establish clinical accuracy or provide a medical diagnosis. The threshold-based method is also limited because physiological responses can differ between individuals, and factors such as sensor placement, movement, and signal quality may affect the readings.

Future work can focus on improving the stress-classification method using machine-learning and deep-learning techniques. Such methods would require a larger dataset with reliable stress labels for proper training and evaluation. Personalized calibration can also be introduced so that the system considers the normal physiological range of individual users instead of applying the same thresholds to everyone.

Longer-term data collection could further help in studying changes in an individual's physiological patterns over time. Improvements to sensor placement, motion-noise handling, and the physical design of the cap may also make the system more suitable for extended use. The IoT interface could additionally be extended to include historical readings, trend visualization, and user-specific monitoring.

Overall, the proposed system demonstrates the feasibility of combining multimodal physiological sensing with IoT-based monitoring in a wearable cap. Larger-scale testing with reliable stress labels and different users will be required to evaluate the classification performance and determine how well the system works under real-world conditions.

REFERENCES

[1] Sattar and M. S. B. Yusoff, "Unveiling the Interplay of Medical Professionalism, Mental Well-being and Coping in Medical Students: a Qualitative Phenomenological Study," BMC Medical Education, Vol. 25, P. 12, 2025, Doi: 10.1186/s12909-024-06595-5.

[2] McDaid, a. L. Vidyasagaran, M. Nasir, S. Walker, J. Wright, K. P. Muliyala, S. Thekkumkara, R. Huque, M. R. Faisal, S. Benkalkar, M. a. Kabir, C. Russell, and N. Siddiqi, "Understanding the Costs and Economic Impact of Mental Disorders in South Asia: a Systematic Review," Asian *Journal of Psychiatry*, vol. 102, p. 104239, 2024, doi: 10.1016/j.ajp.2024.104239.

[3] A. Dewi, R. Thinakan, M. Batumalay, and T. B. Kurniawan, "a Model for Pervasive Computing and Wearable Devices for Sustainable Healthcare Applications," International Journal of Advanced Computer Science and Applications, Vol. 14, No. 10, 2023.

[4] Badr, U. Tariq, F. Al-Shargie, F. Babiloni, F. Al Mughairbi, and H. AlNashash, "a Review on Evaluating Mental Stress by Deep Learning Using EEG Signals," Neural Computing and Applications, Vol. 36, Pp. 12629– 12654, 2024, Doi: 10.1007/s00521-024-09809-5.

[5] Bolpagni, S. Pardini, M. Dianti, and S. Gabrielli, "Personalized Stress Detection Using Biosignals from Wearables: a Scoping Review," Sensors, Vol. 24, no. 10, p. 3221, 2024, doi: 10.3390/s24103221.

[6] Riedl, K. Kostoglou, S. C. Wriessnegger, Et Al., "Videoconference Fatigue from a Neurophysiological Perspective: Experimental Evidence Based on Electroencephalography (EEG) and Electrocardiography (ECG)," Scientific Reports, Vol. 13, P. 18371, 2023, Doi: 10.1038/s41598-02345374-y.

[7] Wang, J. Chen, and J. Ma, "Monitoring and Evaluating the Status and Behaviour of Construction Workers Using Wearable Sensing Technologies," Automation in Construction, Vol. 165, P. 105555, 2024.

[8] Pradhan, S. Bhattacharyya, and K. Pal, "IoT-based Applications in Healthcare Devices," Journal of Healthcare Engineering, Vol. 2021, P. 6632599, 2021.

[9] Blynk, "Blynk IoT platform." [Online]. Available: https://blynk.io/

[10] J. A. Castro-García, *et al.*, "Towards human stress and activity recognition: A review and a first approach based on low-cost wearables," *Electronics*, vol. 11, no. 1, p. 155, 2022.

[11] Empatica, "E4 research wearable sensor platform." [Online]. Available: https://www.empatica.com/research/e4/

[12] G. Vos, K. Trinh, Z. Sarnyai, and M. Rahimi Azghadi, "Generalizable machine learning for stress monitoring from wearable devices: A systematic literature review," *International Journal of Medical Informatics*, vol. 173, p. 105026, 2023, doi: 10.1016/j.ijmedinf.2023.105026.

[13] S. Shanbhog M. and J. Medikonda, "A clinical and technical methodological review on stress detection and sleep quality prediction in an academic environment," *Computer Methods and Programs in Biomedicine*, vol. 235, p. 107521, 2023, doi: 10.1016/j.cmpb.2023.107521.

[14] A. Abd-Alrazaq, M. Alajlani, R. Ahmad, R. AlSaad, S. Aziz, A. Ahmed, M. Alsahli, R. Damseh, and J. Sheikh, "The Performance of Wearable AI in Detecting Stress Among Students: Systematic Review and Meta-Analysis," *Journal of Medical Internet Research*, vol. 26, p. e52622, 2024, doi: 10.2196/52622.

[15] M. Abd Al-Alim, R. Mubarak, N. M. Salem, and I. Sadek, "A machine-learning approach for stress detection using wearable sensors in free-living environments," *Computers in Biology and Medicine*, vol. 179, p. 108918, 2024, doi: 10.1016/j.compbiomed.2024.108918.

[16] H. C. Ates, C. Ates, and C. Dincer, "Stress monitoring with wearable technology and AI," *Nature Electronics*, vol. 7, pp. 98–99, 2024, doi: 10.1038/s41928-024-01128-w.

[17] S. Sosa, A. K. Fontecchio, E. G. Chrysikou, and J. S. Atchison, "Beyond EDA: A Systematic Review of Multimodal Sympathetic Nervous System Arousal Classification for Stress Detection," *Sensors*, vol. 26, no. 5, p. 1584, 2026, doi: 10.3390/s26051584.

[18] D. Rountree-Harrison, S. Berkovsky, and M. Kangas, "Heart and brain traumatic stress biomarker analysis with and without machine learning: A scoping review," *International Journal of Psychophysiology*, vol. 185, pp. 27–49, 2023, doi: 10.1016/j.ijpsycho.2023.01.009.

[19] A. O. Pataca, E. Zdravevski, P. J. Coelho, N. M. Garcia, M. Deryuck, C. Albuquerque, and I. M. Pires, "Use of machine learning for predicting stress episodes based on wearable sensor data: A systematic review," *Computers in Biology and Medicine*, vol. 198, p. 111166, 2025, doi: 10.1016/j.compbiomed.2025.111166.

A Hybrid Deep Learning Framework for Scientific Correction Text Classification

Garima Sharma*, Vikas Tripathi, Vijay Singh
Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India

***Abstract*—Ensuring research integrity and maintaining the reliability of scientific communication requires a clear and comprehensive understanding of the underlying causes of article correction reasons across domains. Correction notices issued to authors are often unstructured with inconsistent textual information, making their automatic classification a challenging task. Existing approaches primarily rely on manual analysis or conventional machine learning techniques that fail to capture the contextual semantics and complex linguistic patterns present in the given correction statements. To address this limitation, this study proposes a deep learning-based framework for the automated classification of correction reasons using a large-scale dataset. The approach integrates contextual semantic representations with a hybrid classification strategy that combines data normalization, ensemble learning, and soft voting mechanisms to enhance classification performance and overall predictive stability. The experimental results demonstrate that the framework outperforms conventional machine learning and existing deep learning algorithms across multiple evaluation metrics. The model achieves an accuracy of 94.3%, along with strong specificity and predictive reliability. The model consistently demonstrates strong evaluation metrics, such as precision, recall, and F1-score, showcasing its effectiveness in handling a complex and imbalanced retraction dataset. The experimental analysis was conducted on a dataset categorized into four distinct classes, namely Correction, Expression of Concern, Reinstatement, and Retraction. The findings of the study offer a scalable and robust solution for systematically identifying correction reasons and uncovering underlying patterns. By enabling transparent, consistent, and efficient interpretation of correction information, this proposed approach can support publishers, academic institutions, policymakers, and researchers in detecting the patterns of scientific misconduct and thus facilitating informed decision-making while mitigating the risk of misinformation in scientific communication.**

Keywords*—Correction analysis; research integrity; BERT; natural language processing; text classification

I. INTRODUCTION

Scholarly publications guarantee the reliability of scientific research. Ensuring the accuracy and credibility of their findings is critical due to their direct and exponential impact on the community health sector [1]. Journals and Publishers employ several mechanisms such as issuing errata for minor errors, expressing concerns, and, in more severe cases, retracting the article to maintain the integrity of the scientific record [2]. A retraction serves to alert readers to articles with significant flaws or erroneous data, rendering their conclusions and findings untrustworthy [3]. These can occur for a variety of reasons, such as fraud, methodological errors, data set and results issues, plagiarism, and publisher errors, as indicated [4]. Although sparsely observed, the rate of withdrawals has increased in recent decades. Even after being retracted, these articles continue to be cited and included in meta-analyses, as evidenced [5], potentially compromising the integrity of the findings in the cited articles. However, no studies have specifically investigated withdrawals in the literature [6]. Authors in [7] aimed to explore the frequency and features of the withdrawn articles investigated and to evaluate the extent to which these retracted publications are cited. Thus, the author who submits manuscripts has the responsibility to ensure that references in their bibliography are not retracted. If an author (or reader) notices that a published article cites a retracted article, they should take the initiative to notify the journal for a correction or withdrawal [8]. The rate and magnitude of withdrawals serve as key indicators of overall operations within the domain of scientific and technical publications, highlighting the need for detailed studies [9]. In recent years, withdrawals have become a significant area of research both nationally and internationally [10]. Three key trends that have emerged in withdrawal patterns are: 1) the rate of retracted published research is growing rapidly, supported by substantial evidence [11]; 2) withdrawals are no longer limited to researchers from developed countries, but have also been extended to developing countries across academic disciplines [12]; and 3) factors of manuscript withdrawal can be grouped into two broad categories: data misconduct and unintentional or honest errors [13].

The integrity of a scholarly publication requires an effective mechanism to detect and remove falsified publications. Recently, there has been a significant increase in the withdrawal of published articles, bringing immediate attention to the reliability of publication practices [14]. These withdrawals support an essential corrective process to prevent the dissemination of false and misleading findings that adversely impact decision-making and scientific progress has been underpinned [15]. In recent years, the reasons for any withdrawal of an article have been too diverse, requiring a systematic classification mechanism. It has been observed that a large body of related research has been carried out using publicly available datasets such as Retraction Watch, which comprise a wide range of factors used to identify an illegitimate publisher [16]. However, with a larger consolidation, the primary reasons are data inconsistency, ethical misconduct, publication malpractices, and erroneous methodologies present, as well as the associated results [17]. Inconsistent, unreliable, and missing data-related information has been found to be dominant in data-related concerns has been observed [18]. In contrast, plagiarism, fabrication, falsification, and, in fact, peer review manipulations

*Corresponding author.

are related to ethical misconduct that undermines scientific credibility [19]. These concerns are widely reported and frequently observed across sensitive domains such as cardiovascular and clinical research, etc., increasing with time has been demonstrated [20].

Thus, the withdrawal of the articles represents a pivotal part of a self-correcting mechanism of science that supports the removal of unreliable, invalid, and ethically compromised findings from scholarly records. This plays a significant role in ensuring academic integrity and transparency of scientific communication in an evidence-based healthcare system. Twitter data to identify issues with publications. They focused on analyzing tweet texts related to three withdrawals of COVID-19 and SARS-CoV-2 publications and their withdrawal notices [21]. Although no early warning signs were found in tweets for one publication, tweets about the other two raised concerns about their validity shortly after publication. The experiments showcased that Twitter could play a role in identifying questionable publications. In 2022, [22] aimed to check how well major search engines, such as Google Scholar, and scientific ones such as PubMed, EMBASE, SCOPUS, and Web of Science, could detect article withdrawal. The retracted articles were taken from the Retraction Watch Database. Although Google Scholar retrieved the majority of the articles, the results revealed that it was not as reliable as scientific databases in identifying the retracted articles. The study recommended using different scientific search engines instead of just a general one. This would help avoid using retracted articles and ensure more reliable research.

To analyze the rate and character of withdrawal articles comprising the literature on infectious diseases and clinical microbiology, [23] conducted a study in which they used the Retraction Watch Database for a cross-sectional analysis of publications retracted up to 30 June 2022 in the said fields. In addition, articles with available citation data from the Web of Science database were also included in the study. The work explored trends in withdrawals and the characteristics of these publications. It also examined the number of references to the articles before and after the withdrawal year. [24] examined the transparency and consistency of the way public health publications were presented to researchers. It was a cross-sectional study that included 441 articles in public health. Data were collated from 11 sources for each publication, with withdrawal notices obtained from publisher websites and full-text aggregators. The criteria of the National Library of Medicine and the Committee on Publication Ethics were used to determine whether an article had been withdrawn. In another work presented [25], data are analyzed on retracted articles to identify trends in withdrawals. Publications were searched in WoSCC and imported into CiteSpace in plain text format for analysis. Through a visual examination of trends in countries, institutions, keywords, and subject areas, the patterns surrounding the retracted articles were revealed. The analysis also highlighted the fields most affected by withdrawals. The comparative overview of recent related work on the classification of withdrawals, highlighting methodological approaches, key findings, and research gaps, is given in Table V (in the Appendix). The existing literature shows significant progress in the identification and categorization of causes for withdrawals. There exists a gap in a universal approach and a computationally robust classification framework. For improving transparency and large-scale analysis, a robust automated system can help address these identified gaps and maintain the integrity of scientific knowledge. In addition, integrative approaches capable of combining domain expertise with data-driven methodological frameworks can help to develop more standardized and scalable classification models. The advancement in artificial intelligence (AI) and its subdomains has recently made it possible to design and develop automated systems that can detect the withdrawal reason of withdrawn articles with high precision and efficiency, as highlighted in [35]. These systems work by analyzing patterns in metadata, textual content, and article history to determine if an article has been retracted and categorize it accordingly [36]. The problem with using retracted literature is that its inclusion in literature reviews can mislead future research and clinical practice. To address these issues, a multi-representation semantic learning framework integrating an enhanced Word2Vec embedding with GloVe and Sentence-BERT to capture lexical, semantic, and contextual information from scientific withdrawal notices has been proposed in this work. These complementary representations are subsequently classified as a hybrid deep learning-based AAHA-ShLNet architecture, offering three key contributions:

- A multi-representation feature extraction framework using Enhanced Word2Vec, GloVe, and Sentence-BERT for scientific withdrawal text.
- Development of a hybrid AAHA-ShLNet classifier integrating AAHA-BN-ShNet, LeNet, and soft voting for robust classification.
- Validation of the proposed framework on 52,811 scientific post-publication notice records using a group-aware partitioning mechanism, demonstrating its effectiveness across multiple evaluation metrics.

II. MATERIALS AND METHODS

The work proposes a hybrid deep learning framework to classify withdrawal-related categories. The framework, as given in Fig. 1, integrates an advanced text representation technique along with a mutually reinforcing neural network architecture to improve the accuracy and robustness of the classification framework. The comprehensive workflow consists of three main sequential phases: 1) Accumulated Data Preprocessing, 2) Key Feature Engineering, and 3) Hybrid Classification.

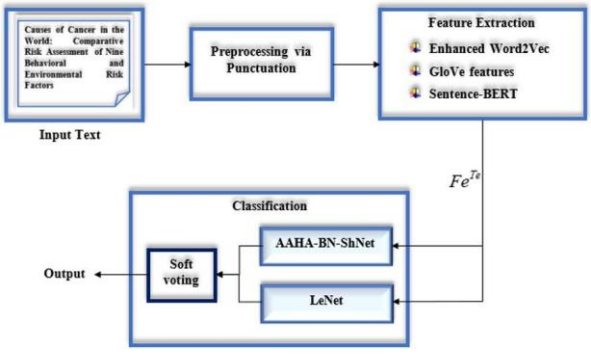


Fig. 1. Workflow architecture of proposed framework

A. Data Preprocessing via Punctuation

1) Dataset and class distribution: The proposed work utilises the four-class post-publication retraction notice records obtained from the Retraction Watch Database for four-class post-publication notice classification. The target variable, Retraction Nature, comprises four editorial-action categories: Correction, Expression of Concern, Reinstatement, and Retraction; the description of each is given in Table. I. After loading the database using an encoding-compatible procedure and validating the record structure, the resulting corpus contained 52,811 records.

TABLE I. DESCRIPTION OF POST PUBLICATION NOTICES

Category	Description
Correction	Published article requires correction, without constituting a full retraction of the article.
Expression of Concern	Concern regarding the reliability, integrity, or validity of a publication.
Reinstatement	Retracted publication has been restored to the active literature databases.
Retraction	Withdrawal of a published article from the scientific record

2) Data preprocessing: Text Processing is used to standardize and clean the collated input data, Tei, prior to feeding the same to the feature extraction engine. Specifically, punctuation has been handled along with the removal of redundant symbols without disturbing meaningful delimiters such as periods and commas. This step also ensures that the sentence boundaries and contextual structure remain intact while reducing the overall noise in the refined dataset. In addition, the output of this stage is punctuated text, TPej, serving as input for the subsequent extraction of features.

B. Feature Extraction

To capture the syntactical and semantic characteristics of the text effectively, a multi-representation feature extraction strategy has been adopted comprising three complementary embedding techniques:

- Enhanced Word2Vec: For dense vector representations of words using CBOW and Skip-Gram architectures as shown in Fig. 2, Word2Vec is utilized. Word2Vec is a shallow neural network model applied to learn word embeddings in NLP.

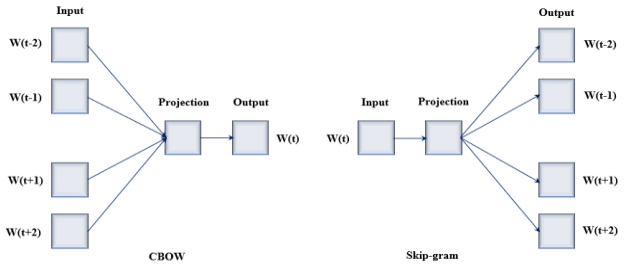


Fig. 2. Word2Vec architectural diagram.

The standard cosine similarity measures the angular relationship between two vectors, keeping the embedding dimensions equal. In scientific publication withdrawal texts, embeddings are often sparse, noisy, and have a highly imbalanced distribution due to domain-specific terminologies and unequal class frequencies. To improve the capability of the used architectures and capture semantic similarity within complex text more effectively, a modified measure of cosine similarity is introduced. This modification incorporates the following:

- An adaptive scaling factor to adjust similarity based on data variability and to reduce the influence of extreme embedding values.
- Log-exp transformation to reduce the influence of extreme values and stabilize semantic distances. This enhancement improves the robustness of the system to capture the semantic relationships across diverse textual distributions. The resulting embeddings are denoted as:

$$cos(\theta) = \{X \cdot Y\}\{/|X||Y|\} \quad (1)$$

The most significant similarity measure utilized by Word2Vec for word vector comparison is cosine similarity, which measures the angle between two vectors as defined by Eq. (1). Increasing the cosine similarity score (towards 1) indicates an increase in similarity among the words or vectors. This measure of similarity can be very important in classifying articles on the basis of some features. However, Word2Vec does not capture the entire similarity between whole sentences or documents. Word2Vec operates by evaluating similarities among word vectors, but when combining these vectors for sentence or document understanding, the process becomes limited. In order to overcome these issues, a modification to the standard cosine similarity measure used by Word2Vec. An adaptive scaling factor δ is used, which adjusts the similarity measure based on the variability of the dataset. This allows the model to adapt to different types of text distributions, making it more robust and accurate, especially in complex text domains. Using the log-exp transformation, the extreme values are smoothed, resulting in a similarity measure that is more balanced and less sensitive to outliers. The updated version of the cosine similarity equation [37], as given and utilized, is defined in Eq. (2).

$$MS(X,Y) = -log\left(1 - exp\left(-\delta \cdot Cos(X,Y)\right)\right) \quad (2)$$

To compute cosine similarity between two vectors X and Y, the dot product is divided by the product of the vectors' magnitudes [38], as given in Eq. (3).

$$cosine(X,Y) = \sum_{i=1}^{n} Xi * Yi / \sqrt{\sum_{i=1}^{n} Xi^2} * \sqrt{\sum_{i=1}^{n} Yi^2} \quad (3)$$

The scaling parameter δ can be calculated using Eq. (4).

$$\delta = \left(Cos(X,Y)\right) - min(Cos(X,Y))/mean(Cos(X,Y)) \quad (4)$$

The four-step process for computing the modified cosine similarity includes:

1) Compute the modified cosine similarity based on the updated expression in Eq. (2).

2) The distance (d) between two vectors X and Y is obtained using Eq. (5).

$$d = 1 - ModifiedSimilarity(X, Y) \quad (5)$$

3) Depending on the distance value d, if $d < 0.5$, then the returned value will be either Eq. (6) or Eq. (7):

$$(1 - d) * (d * d) + d * \sqrt{d} \quad (6)$$

or

$$\sqrt{d} \quad (7)$$

4) Once the final output has been evaluated, the process stops.

By using the updated cosine similarity, Enhanced Word2Vec generates word vectors that represent word meanings and their relationships; it is represented as $W2_{fe}$

- GloVe Word Embedding: The GloVe word embedding method [14] is employed to retrieve semantic features from text, P_{iTe}. GloVe, or Global Vectors, captures words as a vector space. It is simple to train because of its parallel implementation and assists in capturing word relationships. In the matrix Z, Z_{ij} signifies the count of context word i occurring with word j. The GloVe method is designed to minimize the resulting objective function as defined by Eq. (8).

$$G^{fe} = Minimize\Sigma_{i,j=1}^{V}\left(W_i^t W_j + B_i + B_j - logZ_{ij}\right) \quad (8)$$

The output, G^{fe} from GloVe is a word embedding for each word, W, in the input text P_{iTe}.

- Sentence BERT: BERT is a transformer model with an attention mechanism that consumes input text in a cost-effective manner [39]. It is an encoder-only transformer architecture that uses an attention mechanism to process input text and make predictions based on it. BERT processes context in both directions and yields a deeper sense of word meaning. Sentence-BERT creates better sentence-level embeddings, which makes it useful for tasks like retraction classification of published articles by knowing the meaning of entire sentences. BERT's output, Be^{fe} is generally a batch of contextualized word embeddings or sentence embeddings. The overall text-based features are obtained from this feature extraction stage, which employs Enhanced Word2Vec, GloVe, and sentence-BERT to extract features from the punctuated text. The resulting features from this stage are denoted as Fe_{Te}.

$$F_e = [E_{EW2V}||E_{GloVe}||E_{SBERT}] \quad (9)$$

C. Hybrid Classification

After the extraction of text-based features Fe^{Te}, these are used in this classification process. The aim of the classification is to classify retractions. For this, a unique hybrid AAHA-ShLNet strategy is introduced that combines the AAHA-BN-ShNet and LeNet models. The probability outcomes from both classifiers are then combined via soft voting to predict the final classification. Fig. 3 presents a structural depiction of this AAHA-ShLNet strategy.

1) Attention-driven adaptive hybrid activation and batch normalization updated ShuffleNet (AAHA-BN-ShNet): ShuffleNet, a lightweight grouped convolution and channel shuffle operations provide an effective mechanism to learn local feature interactions from dense embedding representation. This approach is being utilized, as the expected inputs are not the raw texts but an embedding matrix. The model aims to achieve maximum efficiency and precision using advanced methods like group convolution (GConv), pointwise group convolution (PWGConv), shuffling, and depthwise convolution (DWConv) [40]. Here, ShuffleNet does not depend on standard "point convolution" in convolution layers but employs GConv, where all the inputs belonging to a group are processed separately. This segregated way ensures that accurate information related to each group is identified, thus enhancing precision. In ShuffleNet, the standard BN normalizes the input layer by layer, which stabilizes the learning process regardless of possible network weight changes. However, it is not necessarily optimized in every layer, and it does not provide smoothness for negative values, which can cause computational inefficiencies or overfitting problems, specifically in article withdrawal reason classification. This overfitting slows down model performance by having the model memorize useless patterns in the data, instead of generalizing well to original inputs.

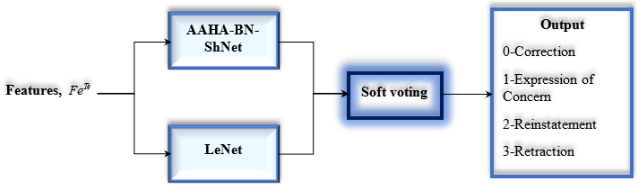


Fig. 3. Structural process of AAHA-ShLNet strategy.

In consideration of such challenges, the ShuffleNet model is improved with the inclusion of an Adaptive Hybrid Activation and Attention-based Batch Normalization (AAHA-BN), which regulates activations in training and enhances the overall performance of the model. Such an enhancement facilitates better generalization of the model through efficient normalization of activations and avoids the susceptibility that comes with traditional BN. The improved ShuffleNet model, tailored specifically for withdrawal text classification in publications, accepts text-based features, FeTe, as input and goes through a series of layers to efficiently classify. The model starts with a convolution (Conv) layer to generate the initial feature extraction of the input text. This is followed by AAHA-BN normalizing the activations to promote stability, followed by the use of LReLU to impose non-linearity with the inclusion of allowing small negative values. A max pooling layer is then used to decrease dimensionality while retaining important features. The model continues to Stage 2 and Stage 4, which involve Conv layers, AAHA-BN [41], and LReLU, enhancing the feature extraction process in stages. In Modified Stage 3, the model continues to improve feature representation with parallel processing paths involving extra DWConv and AAHA-BN before channel shuffling is used to combine results. After these steps, the features are passed through GAP to convolve all the

feature maps together and are then input into an FC layer to generate the final output. The last output of the AAHA-ShLNet model is indicated as OtAAHA-BN-ShNet.

During Modified Stage 3, as indicated in Fig. 4(a) (given in Appendix), the input passes through two parallel paths to boost feature extraction. In the first path, the input initially goes through a Conv layer, followed by the AAHA-BN to normalize activations. The output subsequently passes through an LReLU activation and a DWConv layer, followed by a second AAHA-BN, a Conv layer, and an additional AAHA-BN. Lastly, the output is again activated by LReLU. The second branch uses a less complex flow: it begins with DWConv, then AAHA-BN, Conv, and LReLU activation. Once both branches have been processed, the outputs are concatenated and fed into channel shuffling, which rearranges the features in an efficient manner for the subsequent layer. Fig. 4(b) (given in the Appendix), the input is subjected to channel splitting prior to passing it through a sequence of layers such as a Conv layer, AAHA-BN, LReLU, and DWConv. The output is subsequently processed through AAHA-BN and a Conv layer prior to concatenating with the earlier output. Channel shuffling is the last step in this structure, which efficiently shuffles the features from various channels to enhance feature diversity.

The input is initially divided into several channels for parallel processing to enable the model to learn more varied features for both Stage 2 and Stage 4. The input goes through a Conv layer, followed by BN and a ReLU function. The model then uses a DWConv layer to process the feature representation more precisely, followed by a second BN layer, a Conv layer, and a final BN layer. Then, the result from both the channel split and BN + ReLU branch is concatenated. Then the features undergo channel shuffling to effectively merge the results before proceeding to the next step. The standard BN formula normalizes the activations by adjusting the mean and variance for each feature within a mini-batch, as defined by Eq. (10).

$$\hat{y}_i = \frac{y_i^l - \mu_B^l}{\sqrt{\sigma_B^{l2} + \epsilon}} \tag{10}$$

where, Yi is the layer feature map, denotes a constant, and implies the mean and variance of y^l. In this model, the standard BN is enhanced, and its updated expression is provided in Eq. (11).

$$\hat{y}_i = \frac{y_i^l - \mu_B^l}{\sqrt{\sigma_B^{l2} + \epsilon}} X\ P(y) \tag{11}$$

P(y) signifies the Lp pooling with AAHA-BN-based normalization. This pooling method offers better generalization compared to max pooling. Instead of choosing the maximum value, Lp pooling [42] calculates a weighted average of the input within the pooling region [43].

2) LeNet model: LeNet is a very effective Deep Neural Network (DNN) model that takes very little computational time. It consists of seven layers, including the output layer [44]. The input layer of the network accepts text-based features FeTe, which are then passed to the next layer after applying a size-normalization operation. The convolutional layer is the next layer, which uses a collection of feature maps or kernels to select key features of the input. These kernels are collections of square matrices, all with the same weights. The convolutional layer applies learnable kernels over the structured numerical feature representation to identify local feature interactions. In the proposed framework, the input is not an image; rather, the fused textual embeddings are arranged as a fixed-dimensional numerical feature map. The convolution operation therefore enables the LeNet model to extract informative local patterns from the structured embedding representation.

3) Final classification using soft voting: Following the withdrawal classification by both AAHA-BN-ShNet and LeNet classifiers [45] on articles, the probabilities from both classifiers' outputs are used as inputs to the soft voting to get the final classification. Soft voting works by evaluating the weighted sum of the prediction probabilities for each class from two classifiers, as expressed in Eq. (12). The class with the highest overall probability is then chosen as the final prediction.

$$\hat{X}_i = argmax_l \sum_{j=1}^{n} \omega_j P_{i,l}^j \tag{12}$$

Here, j^{th} classifier weight is implied; the number of classifiers is represented by n=2, and is the probability predicted by the jth classifier for the ith sample to be classified as the lth class. By combining the AAHA-BN-ShNet and LeNet classifiers, the proposed hybrid AAHA-ShLNet strategy successfully categorizes withdrawal reasons of articles, which labels the outcome as 0, 1, 2, and 3 (Correction, Expression of Concern, Reinstatement, Retraction).

III. RESULTS

The proposed framework was simulated using PYTHON 3.7. The processor employed was "11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz and the Installed RAM size was 16.0 GB. Moreover, the analysis of withdrawal reason classification on articles was conducted using the Retraction Watch Database from its GitHub repository [46]. This Database aids as a complete repository of retractions—instances where available research articles were withdrawn due to numerous reasons, including "fraud, plagiarism, or ethical concerns." The record contains 52,811 entries, the newest retraction repository. For the experiments, a complete retraction dataset of 52,811 records was utilized, categorized into four diverse classes: Correction (Label 0), Expression of Concern (Label 1), Reinstatement (Label 2), and Retraction (Label 3). The class distribution is highly imbalanced. Retraction constitutes 48,335 records (91.52%), followed by Expression of Concern with 3,092 records (5.85%), Correction with 1,245 records (2.36%), and Reinstatement with 139 records (0.26%). Thus, the minority Reinstatement category contains substantially fewer observations than the majority Retraction category. Because overall accuracy can therefore be dominated by the majority class, macro-averaged F1-score, balanced accuracy, class-wise precision, recall, F1-score, and Matthews correlation coefficient (MCC) *are* additionally considered in the evaluation.

A. Performance Evaluation

To assess the performance of the framework introduced for the classification of withdrawal articles, a diverse set of performance metrics is used. The selected metrics provide a

comprehensive and balanced evaluation of classification capacity, especially in unbalanced class scenarios. Let T_p denote true positives, T_n be true negatives, F_p false positives, and F_n be false negatives.

1) Accuracy: Accuracy measures the general correctness of the classification model as given in Eq. (13).

$$Accuracy = \frac{T_p+T_n}{T_p+T_n+F_p+F_n} \quad (13)$$

Although accuracy provides a general indication of performance, it may be misleading when applied to imbalanced datasets.

2) Precision: Precision measures the accuracy of predicted positive instances using Eq. (14).

$$Precision = \frac{T_p}{T_p+F_p} \quad (14)$$

3) Recall: Recall, also called sensitivity, is an indicator to accurately measure the effectiveness of the model to identify positive instances, as given in Eq. (15).

$$Recall = \frac{T_p}{T_p+F_n} \quad (15)$$

4) Specificity: Specificity measures a model's ability, as per Eq. (16), to correctly identify the negative instances.

$$Specificity = \frac{T_n}{T_n+F_p} \quad (16)$$

The higher value of specificity ensures that fewer incorrect classifications of non-withdrawal cases have been identified.

5) F1-Score: The F1-score in Eq. (17) supports a value to be used as a balance between precision and recall:

$$F1-score = \frac{2*T_p}{2*T_p+F_p+F_n} \quad (17)$$

It is a crucial value for unbalanced datasets.

6) Matthews Correlation Coefficient (MCC): As per the MCC value, Eq. (18) provides a balanced measure for considering all elements of the confusion matrix:

$$MCC = \frac{T_p*T_n-F_pF_n}{\sqrt{(T_p+F_p)(T_p+F_n)(T_n+F_p)(T_n+F_n)}} \quad (18)$$

The MCC value ranges from-1(total disagreement) to +1 (perfect prediction).

7) Negative Predicted Value (NPV): The NPV in Eq. (19) helps to measure the proportion of correctly predicted negative instances:

$$NPV = \frac{T_n}{T_n+F_n} \quad (19)$$

8) False Positive Rate (FPR): The FPR value given in Eq. (20) helps quantify the proportion of negative instances incorrectly classified as positive:

$$FPR = \frac{F_p}{F_p+T_n} \quad (20)$$

Lower FPR values indicate reduced false alarms.

9) False Negative Rate (FNR): The FNR value shows the exact proportion of all positive instances incorrectly classified as negative instances, calculated using Eq. (21):

$$FNR = \frac{F_n}{F_n+T_p} \quad (21)$$

Minimizing FNR is critical to avoid missing relevant withdrawal cases.

For rigorous validation, the proposed framework is evaluated using multiple training ratios and compared with baseline models. In addition to this, statistical measures including mean, median, standard deviation, and range, together with ablation analysis, are used to assess the robustness and reliability of the framework.

B. Comparative Performance Analysis

To evaluate the efficacy of the proposed model, a comparative analysis was conducted against several established models, including CNN, Bi-LSTM, GoogLeNet, Bi-GRU, LSTM, RNN, and AlexNet. The results showed that the model achieved superior performance in four-class post-publication notice classification, achieving an accuracy of 94.3%. In contrast, baseline models exhibit comparatively lower accuracy levels. Furthermore, in 90% of the training data, AAHA-ShLNet achieves a precision score of 0.885, highlighting its effectiveness. Baseline models report lower precision values, including CNN (0.836), Bi-LSTM (0.819), GoogLeNet (0.848), Bi-GRU (0.839), LSTM (0.859), RNN (0.826), and AlexNet (0.829).

The Negative Predictive Value (NPV) analysis demonstrates the robustness of the proposed model in identifying and classifying the removed articles. The model achieved the highest NPV of 0.927 in 60% of the training data, indicating a strong predictive capacity. The model was further improved from 0.936 to 0.943 with an increase in data from 70% to 90%, respectively. Hence, these results confirm the consistent enhancement of classification reliability with an increase in the dataset.

The effective withdrawal of categorization requires minimal negative performance indicators, such as false positive rates (FPR). The proposed model achieved the lowest FPR value of 0.032 in the training data of 90%, demonstrating a significant reduction in errors. In contrast, conventional models show higher FPR scores with Bi-LSTM (0.063), RNN (0.061), AlexNet (0.060), CNN (0.057), and Bi-GRU (0.059). The results highlight the proposed model's competence in minimizing classification errors. In addition, the Word2Vec component integrated an adaptive scaling factor and log-exp transformation to regulate similarity measures according to the variability of the dataset, thus producing a more balanced similarity measure less sensitive to outliers. The detailed description of all performance metrics compared to the traditional system is given in Table II. The statistical analysis is expanded to include macro-averaged precision, recall, and F1-score, balanced accuracy, class-wise precision/recall/F1, and a four-class confusion matrix. The performance of the minority Reinstatement class is explicitly reported rather than being represented only through aggregate accuracy.

TABLE II. COMPARATIVE PERFORMANCE ANALYSIS OF PROPOSED AND BASELINE MODELS

Model	Accuracy	Precision	FPR	FNR
AAHA - ShLNet	0.943	0.885	0.032	0.157
CNN	0.920	0.836	0.057	0.180
Bi-LSTM	0.913	0.819	0.063	0.195
GoogLeNet	0.920	0.848	NR	0.180
Bi-GRU	0.915	0.839	0.059	0.190
LSTM	0.919	0.859	NR	0.175
RNN	0.915	0.826	0.061	0.190
AlexNet	0.915	0.829	0.060	NR

C. Statistical Evaluation

To assess the effectiveness of the framework, a comprehensive quantitative evaluation was conducted using statistical measures of accuracy. The results, summarized in Table III, provide a comparative analysis of the introduced model compared to several advanced deep learning models and the baseline, such as CNN, LSTM, Bi-LSTM, GoogLeNet, Bi-GRU, RNN, and AlexNet. Statistical evaluation includes minimum, mean, standard deviation, median, and maximum accuracy values. Thus, a comprehensive understanding of the performance of the model under different experimental conditions is possible. Among all evaluated models, the proposed model demonstrates superior and consistent performance. More specifically, the mean accuracy for the proposed architecture was estimated at 0.943, which is higher than the mean accuracy of all baseline models, ranging from 0.913 to 0.920. It should be noted that the model demonstrates an ability to achieve fairly stable prediction performance across multiple iterations and datasets. In addition, the maximum accuracy of 0.972 provided by the proposed model outperforms all other baselines, including LSTM (0.950), GoogLeNet (0.946), and CNN (0.944). Thus, the suggested architecture allows for the highest classification performance under optimal conditions. As for the model's robustness, its relatively small standard deviation equals 0.019, which signifies that the performance remains quite stable and there are no major differences between different experiments. The median accuracy of 0.941 supports this statement, as it almost coincides with the mean value. The baseline models not only have a lower average accuracy than the proposed model, but also show a wider range of performance, as shown by their higher or similar standard deviation values. This means that traditional architectures may have trouble generalizing well when there are complex and unbalanced withdrawal data. Combining multi-level feature representations makes the model better at capturing both semantic and contextual information. In addition, using Adaptive Hybrid Activation with Attention-based Batch Normalization (AAHA-BN) improves feature normalization and makes training more stable. The hybrid classification strategy that combines AAHA-BN-ShNet and LeNet also helps the decision to be more general and stronger. In general, statistical analysis confirms that the proposed framework achieves higher precision, improved stability, and superior generalization capability compared to existing approaches, making it highly suitable for reliable four-class post-publication notice classification in the scientific literature.

TABLE III. STATISTICAL EVALUATION OF ACCURACY

Model	Min	Mean	SD	Median	Max
AAHA-ShLNet	0.919	0.943	0.019	0.941	0.972
CNN	0.891	0.920	0.018	0.920	0.944
LSTM	0.891	0.919	0.022	0.918	0.950
Bi-LSTM	0.884	0.913	0.020	0.916	0.938
GoogLeNet	0.896	0.920	0.019	0.919	0.946
Bi-GRU	0.887	0.915	0.021	0.916	0.943
RNN	0.889	0.915	0.019	0.916	0.939
AlexNet	0.888	0.915	0.019	0.916	0.940
TF-IDF + Linear SVM	0.753	0.86	0.052	0.855	0.88

To examine the effectiveness of the proposed model, a comparison is made with two distinct variations, one with enhanced ShuffleNet and the other without enhanced Word2vec. The details of these comparisons are summarized in Table IV. The results of the ablation evaluation clearly highlight the superior performance of the AAHA-ShLNet model. Specifically, the scheme achieved the maximum sensitivity rate of 0.843, outperforming AAHA-ShLNet without enhanced ShuffleNet (0.747) and AAHA-ShLNet without enhanced Word2vec (0.699). In addition, the AAHA-ShLNet scheme acquired the lowest FPR score of 0.032, indicating reduced error values. In contrast, the proposed model with existing ShuffleNet and AAHA-ShLNet with existing Word2vec recorded relatively higher FPR values of 0.084 and 0.100, respectively.

TABLE IV. ABLATION STUDY OF THE PROPOSED FRAMEWORK

Measure	AAHA-ShLNet	w/o Enhanced ShuffleNet	w/o Enhanced Word2Vec
Accuracy	0.943	0.916	0.900
MCC	0.791	0.663	0.599
Sensitivity	0.843	0.747	0.699
FNR	0.157	0.253	0.301
Specificity	0.922	0.874	0.850
NPV	0.948	0.916	0.900
Precision	0.885	0.748	0.700
F-Measure	0.843	0.747	0.700
FPR	0.032	0.084	0.100

IV. DISCUSSION

The proposed experimental work provides a comprehensive assessment of four-class post-publication notice classification using multiple training-data configurations and class-sensitive performance measures, as compared to the traditional approaches on multiple evaluation metrics, thus validating its effectiveness for an automated classification of article withdrawal reasons. An accuracy of 94.3% has been achieved, with an improvement across all the key metrics such as precision, recall, specificity, and predictive reliability. The hybrid structure enables the model to effectively manage the complex and imbalanced retraction datasets. This improved performance is likely due to the fusion of multilevel semantic

feature extraction and hybrid classification mechanisms. The withdrawal notices often have inconsistent reporting structures, leading to an ambiguous understanding along with domain-specific terminologies, making the automated interpretation highly challenging. Similar concerns have been highlighted by Rao et al. [22], Hwang [12], and Vuong [30]. The result confirms that the hybrid approach has a better generalization capability than standalone models for handling the heterogeneous textual data.

The convolutional components in the proposed framework do not operate on image pixels or visual content. Instead, they operate on a structured numerical feature map constructed from the fused textual representations. Enhanced Word2Vec, GloVe, and Sentence-BERT generate complementary numerical representations at different levels of linguistic abstraction. These representations are concatenated to form a unified feature vector and subsequently arranged into a fixed-dimensional numerical feature map. The two-dimensional arrangement is therefore a computational representation rather than an assumption that the input possesses visual semantics. Convolutional filters are used to learn local interactions among neighboring embedding dimensions and feature channels. Grouped convolution and channel-shuffling operations in ShuffleNet provide parameter-efficient learning of these interactions, while LeNet provides a complementary convolutional pathway. Thus, the use of convolution is motivated by structured feature interaction learning rather than by treating textual data as an image.

The semantic understanding as well as the contextual representation has improved by integrating advanced embedding techniques like Enhanced Word2Vec, GloVe, and Sentence-BERT. On one side, the traditional approaches often fail to capture long-range contextual dependencies and their semantic relationships, while on the other hand, the proposed multi-representation embedding strategy enabled the model to capture the linguistic patterns at the word, sentence, and even contextual level simultaneously. The resulting tensor is a structured feature representation rather than a natural image, and the convolutional layers are employed as computational feature extractors over locally arranged embedding dimensions. The overall robustness of the model against noisy and unstructured text has been improved by utilizing modified cosine similarity with enhanced word2Vec by reducing the sensitivity towards outliers and variable data distribution. A similar observation has been demonstrated by Yusof et al. [37], Shepperd & Yousefi [34], and Rao et al. [22]. Therefore, this fusion has substantially improved the classification reliability for complex scientific datasets comprising ambiguous textual patterns.

The incorporation of Adaptive Hybrid Activation with Attention-based Batch Normalization (AAHA-BN) and the hybrid soft voting classification strategy enhanced the feature normalization and stabilized the learning process of the model by effectively regulating the activation distribution during the training process, which is another important contribution of this study. This combination supports the reduction of overfitting issues and improvement in convergence stability while processing imbalanced datasets. For integrating hierarchical feature extraction adaptability with computational efficiency, the hybrid combination of AAHA-BN-ShNet and LeNet was found to be effective. The predictive robustness has been enhanced by the aggregation of complementary classifier outputs into a unified decision-making framework using a soft voting mechanism. Further, the ablation study performed has clearly demonstrated a noticeable degradation in classification metrics such as accuracy, sensitivity, and specificity in the case of removing any of the above mechanisms from the model. A similar discussion has been presented by Baker [28] that shows the significance of normalization and robust features in deep learning approaches defined by Sharma et al. [36] and Chan et al. [32]. Hence, this collective approach is best suited for reliable classification tasks.

The proposed framework thus offers a scalable and efficient solution for automatic scientific integrity monitoring and identification of unreliable publications across multiple research domains. The recent gradual hike in article withdrawals globally highlights the urgent requirement for such intelligent systems that are capable of supporting publishers, indexing agencies, policymakers, and researchers in identifying the root cause of withdrawal more effectively. This systematic approach helps reduce the propagation of misinformation and further improves transparency in scientific communications. Haunschild & Bornmann [23] and Pastor-Ramón et al. [24] have highlighted similar concerns in their studies and emphasized the requirement of such a cohesive and robust system. With these promising results, there are certain limitations with this work. First, the study heavily relies on the Retraction Watch Database, and thus the completeness, terminologies, and reporting practices of post-publication notices may vary from publishers, journals, as well as disciplines. Consequently, bias may occur depending on the data source. Second, the existing data imbalance estimation of minority-class performance remains more difficult when class support is limited. Third, temporal or domain shifts can impact the overall result and may therefore require additional validation on future records and datasets from independent sources. Fourth, a requirement for some leakage control mechanism to avoid similar records in analysis. Future work may include utilizing explainable AI techniques for supporting multilingual withdrawal notices along with multi-label classification to further improve the scalability and real-world applicability for automatic article withdrawal analysis. External validation across independent databases and multilingual or cross-publisher collections is therefore an important direction for future work.

V. CONCLUSION

The work presents a novel hybrid framework, termed AAHA-ShLNet, for the four-class classification of post-publication editorial notices. The proposed approach integrates a structured pipeline consisting of text preprocessing, multi-level feature extraction, and hybrid classification. First, data preprocessing is performed to remove redundant punctuation while keeping the essential linguistic structure. In addition, semantic and contextual representations are captured using enhanced embedding techniques such as Enhanced Word2Vec, GloVe, and Sentence-BERT, allowing exhaustive feature learning from collated textual data. For classification, a hybrid strategy is employed that combines AAHA-BN-ShNet and LeNet, where the outputs of both models are aggregated using a soft voting mechanism to improve predictive robustness. The experimental results show that the proposed framework is competitive with conventional models. In particular, the model

achieved a precision of 0.885, better than the traditional baseline approaches such as CNN(0.836), Bi-LSTM(0.819), GoogLeNet(0.848), Bi-GRU(0.839), LSTM(0.859), RNN(0.826), and AlexNet (0.829). Performance gains are attributed to the integration of complementary feature representations and the enhanced normalization strategy incorporated within the AAHA-ShLNet architecture. Furthermore, the hybrid decision fusion mechanism contributes to improved generalization and reduced classification errors. In general, the proposed framework provides an effective and scalable solution to identify withdrawal-related patterns. By allowing for more accurate detection of unreliable publications, the model supports informed decision-making, reduces the risk of spreading misinformation, and contributes to maintaining the integrity and reliability of scientific communication. In future work, the framework will be extended to support multi-label and fine-grained classification scenarios where multiple underlying causes can be associated with a single article. Furthermore, explainable artificial intelligence (XAI) techniques can be integrated to improve the interpretations and transparency of improved classification decisions. The framework demonstrates strong potential for real-world deployment in large-scale scientific monitoring systems.

ACKNOWLEDGMENT

The authors would like to acknowledge their university for providing the infrastructure required to carry out the experiments.

DECLARATION OF GENERATIVE AI USAGE

During the preparation of this manuscript, no generative artificial intelligence tools were used.

REFERENCES

[1] B. A. Nosek, D. B. Allison, K. H. Jamieson, M. McNutt, A. B. Nielsen, and S. M. Wolf, "A framework for assessing the trustworthiness of scientific research findings," *Proceedings of the National Academy of Sciences*, vol. 123, no. 6, Art. no. e2536736123, 2026.

[2] Z. Syed, F. Syed, L. Thabane, and M. Rodrigues, "COVID-19 retracted publications on Retraction Watch: A systematic survey of their preprints and citations," *Heliyon*, vol. 9, no. 4, 2023.

[3] I. Campos-Varela and A. Ruano-Raviña, "Misconduct as the main cause for retraction: A descriptive study of retracted publications and their authors," *Gaceta Sanitaria*, vol. 33, pp. 356–360, 2019.

[4] X. Zheng, J. Chen, A. Tollas, and C. Ni, "The effectiveness of peer review in identifying issues leading to retractions," *Journal of Informetrics*, vol. 17, no. 3, Art. no. 101423, 2023.

[5] Q. H. Vuong, V. P. La, M. T. Ho, T. T. Vuong, and M. T. Ho, "Characteristics of retracted articles based on retraction data from online sources through February 2019," *Science Editing*, vol. 7, no. 1, pp. 34–44, 2020.

[6] C. Chen, Z. Hu, J. Milbank, and T. Schultz, "A visual analytic study of retracted articles in scientific literature," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 2, pp. 234–253, 2013.

[7] X. Liu, C. Wang, D. Z. Chen, and M. H. Huang, "Exploring perception of retraction based on mentioned status in post-retraction citations," *Journal of Informetrics*, vol. 16, no. 3, Art. no. 101304, 2022.

[8] F. C. Fang, R. G. Steen, and A. Casadevall, "Misconduct accounts for the majority of retracted scientific publications," *Proceedings of the National Academy of Sciences*, vol. 109, no. 42, pp. 17028–17033, 2012.

[9] C. Candal-Pedreira, C. Guerra-Tort, A. Ruano-Ravina, F. Freijedo-Fariñas, J. Rey-Brandariz, J. S. Ross, and M. Pérez-Ríos, "Retracted papers originating from paper mills: A cross-sectional analysis of references and citations," *Journal of Clinical Epidemiology*, vol. 172, Art. no. 111397, 2024.

[10] C. Brook, N. Lynøe, A. Eriksson, and D. Balding, "Retraction of a peer-reviewed article suggests ongoing problems with Australian forensic science," *Forensic Science International: Synergy*, vol. 3, Art. no. 100208, 2021.

[11] E. Sung, J. L. Ramirez, A. Z. Becerra, E. Jaramillo, W. J. Gasper, P. Schneider, M. S. Conte, and J. C. Iannuzzi, "Identifying disruptive publications and changing paradigms in vascular surgery," *JVS–Vascular Insights*, vol. 2, Art. no. 100036, 2024.

[12] S. Y. Hwang, D. K. Yon, S. W. Lee, M. S. Kim, J. Y. Kim, L. Smith, A. Koyanagi, M. Solmi, A. F. Carvalho, E. Kim, and J. I. Shin, "Causes for retraction in the biomedical literature: A systematic review of studies of retraction notices," *Journal of Korean Medical Science*, vol. 38, no. 41, Art. no. e333, 2023.

[13] P. Sebo and M. Sebo, "Comparing the performance of Retraction Watch Database, PubMed, and Web of Science in identifying retracted publications in medicine," *Accountability in Research*, vol. 33, no. 2, 2026.

[14] B. F. Kocyigit, A. Zhaksylyk, A. Akyol, and M. Yessirkepov, "Characteristics of retracted publications from Kazakhstan: An analysis using the Retraction Watch Database," *Journal of Korean Medical Science*, vol. 38, no. 46, Art. no. e390, 2023.

[15] P. Khurana, K. Sharma, and Z. Uddin, "Unraveling retraction dynamics in COVID-19 research: Patterns, reasons, and implications," *Accountability in Research*, vol. 32, no. 7, pp. 1241–1264, 2025.

[16] M. P. Contreras-Muñoz, C. Zahn-Muñoz, E. Solís-Albanese, and E. Martínez-Rojas, "Analysis of retractions in nursing from publications between 2000 and 2024: A bibliometric analysis using Retraction Watch," *Nursing Reports*, vol. 15, no. 10, Art. no. 349, 2025.

[17] X. Rossello, S. Raposeiras-Roubín, R. Latini, A. Domínguez-Rodríguez, J. A. Barrabés, P. L. Sánchez, M. Anguita, F. Fernández-Vázquez, D. Pascual-Figal, J. M. de la Torre Hernández, and S. Ferraro, "Rationale and design of the pragmatic clinical trial tREatment with Beta-blockers after myOcardial infarction withOut reduced ejection fracTion (REBOOT)," *European Heart Journal–Cardiovascular Pharmacotherapy*, vol. 8, no. 3, pp. 291–301, 2022.

[18] R. A. Okyay, B. F. Kocyigit, A. B. Qumar, M. Yessirkepov, and H. E. Sumbul, "Fifty years of retracted medical publications from 1975 to 2024: A comprehensive analysis of trends, reasons, and countries using the Retraction Watch Database," *Journal of Korean Medical Science*, vol. 40, no. 46, Art. no. e300, 2025.

[19] J. I. de Granda-Orive, C. A. Jiménez-Ruiz, R. Aleixandre-Benavent, D. López-Padilla, C. Rábade-Castedo, M. J. Gómez, and A. Alonso-Arroyo, "Trends and characteristics of retracted articles in the smoking field: An observational study," *Open Respiratory Archives*, Art. no. 100496, 2025.

[20] P. Sharma, B. Sharma, A. Reza, K. K. Inampudi, and R. K. Dhamija, "A systematic review of retractions in biomedical research publications: Reasons for retractions and their citations in Indian affiliations," *Humanities and Social Sciences Communications*, vol. 10, no. 1, Art. no. 597, 2023.

[21] V. Bhat, S. Kaushik, A. Kumar, S. Bellur, and A. Kalra, "Retractions and withdrawals in clinical cardiovascular literature: Analysis of the Retraction Watch Database," *JACC: Advances*, vol. 4, no. 8, Art. no. 101802, 2025.

[22] D. Rao, J. Young, T. Dietterich, and C. Callison-Burch, "WithdrarXiv: A large-scale dataset for retraction study," 2024.

[23] R. Haunschild and L. Bornmann, "Can tweets be used to detect problems early with scientific papers? A case study of three retracted COVID-19/SARS-CoV-2 papers," *Scientometrics*, vol. 126, no. 6, pp. 5181–5199, 2021.

[24] E. Pastor-Ramón, I. Herrera-Peco, O. Agirre, M. García-Puente, and J. M. Morán, "Improving the reliability of literature reviews: Detection of retracted articles through academic search engines," *European Journal of Investigation in Health, Psychology and Education*, vol. 12, no. 5, pp. 458–464, 2022.

[25] T. Zilberman, I. Margalit, D. Yahav, and N. Tau, "Retracted publications in infectious diseases and clinical microbiology literature: An analysis using the Retraction Watch Database," *Clinical Microbiology and Infection*, vol. 29, no. 11, pp. 1454.e1–1454.e8, 2023.

[26] C. J. Bakker, E. E. Reardon, S. J. Brown, N. Theis-Mahon, S. Schroter, L. Bouter, and M. P. Zeegers, "Identification of retracted publications and completeness of retraction notices in public health," *Journal of Clinical Epidemiology*, vol. 173, Art. no. 111427, 2024.

[27] Q. Qi, J. Huang, Y. Wu, Y. Pan, J. Zhuang, and X. Yang, "Recent trends: Retractions of articles in the oncology field," *Heliyon*, vol. 10, no. 12, Art. no. e33007, 2024.

[28] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, vol. 533, no. 7604, pp. 452–454, 2016.

[29] J. P. A. Ioannidis, J. Baas, R. Klavans, and K. W. Boyack, "A standardized citation metrics author database annotated for scientific field," *PLOS Biology*, vol. 17, no. 8, Art. no. e3000384, 2019.

[30] Q. H. Vuong, "Reform retractions to make them more transparent," *Nature*, vol. 582, no. 7811, p. 149, 2020.

[31] N. S. L. Yeo-Teh and B. L. Tang, "An alarming retraction rate for scientific publications on Coronavirus Disease 2019 (COVID-19)," *Accountability in Research*, vol. 28, no. 1, pp. 47–53, 2021.

[32] A. W. Chan, F. Song, A. Vickers, T. Jefferson, K. Dickersin, P. C. Gøtzsche, H. M. Krumholz, D. Ghersi, and H. B. van der Worp, "Increasing value and reducing waste: Addressing inaccessible research," *The Lancet*, vol. 383, no. 9913, pp. 257–266, 2014.

[33] N. Mascato Fontaíña, C. Candal-Pedreira, G. García, J. S. Ross, A. Ruano-Ravina, and L. Martin-Gisbert, "Identifying common patterns in journals that retracted papers from paper mills: A cross-sectional study," Research Integrity and Peer Review, vol. 10, no. 1, Art. no. 21, 2025

[34] M. Shepperd and L. Yousefi, "An analysis of retracted papers in computer science," *PLOS ONE*, vol. 18, no. 5, Art. no. e0285383, 2023.

[35] J. P. Tennant, F. Waldner, D. C. Jacques, P. Masuzzo, L. B. Collister, and C. H. Hartgerink, "The academic, economic and societal impacts of open access: An evidence-based review," *F1000Research*, vol. 5, Art. no. 632, 2016.

[36] K. Sharma and P. Khurana, "Retracted citations and self-citations in retracted publications: A comparative study of plagiarism and fake peer review," 2025.

[37] N. H. Mohd Yusof, N. A. Mohd Subha, N. Hamzah, S. Buyamin, and M. A. Shamsudin, "Logical entity weighting and modified cosine similarity for automated triples extraction via HMM-BERT in knowledge-based maintenance systems for the cement industry," 2025.

[38] L. Sahu and B. R. Mohan, "An improved K-means algorithm using modified cosine distance measure for document clustering using Mahout with Hadoop," in *Proc. International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2014, pp. 1–5.

[39] A. A. Jamjoom, H. Karamti, M. Umer, S. Alsubai, T. H. Kim, and I. Ashraf, "RoBERTaNet: Enhanced RoBERTa transformer-based model for cyberbullying detection with GloVe features," *IEEE Access*, vol. 12, pp. 58950–58959, 2024.

[40] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6848–6856.

[41] A. A. Ziaee and E. Çano, "Batch Layer Normalization: A New Normalization Layer for CNNs and RNNs," in Proceedings of the 6th International Conference on Advances in Artificial Intelligence (ICAAI '22), Birmingham, U.K., Oct. 2022, pp. 40–49. doi: 10.1145/3571560.3571566.

[42] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks: A Review," CoRR, vol. abs/2009.07485, 2020

[43] C C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," CoRR, vol. abs/1811.03378, 2018.

[44] H. Gholamalinezhad, H. Khosravi, "An improved LeNet deep neural network model for Alzheimer's disease classification using brain magnetic resonance images," *IEEE Access*, vol. 9, pp. 161194–161207, 2021.

[45] K. Akyol, E. Uçar, Ü. Atila, and M. Uçar, "An ensemble approach for classification of tympanic membrane conditions using soft voting classifier," Multimedia Tools and Applications, vol. 83, no. 32, pp. 77809–77830, 2024.

[46] *Retraction Watch Database*. Retraction Watch, Center for Scientific Integrity, 2026. Available: https://retractionwatch.com/retracted-paper-database/.

APPENDIX

TABLE V. COMPARATIVE ANALYSIS OF RECENT STUDIES

Scope	Methodology	Findings	Identified Gaps
Life Science Journals [8]	Retrospective Analysis	delay between publication and withdrawal	Delayed detection mechanisms
Engineering and Applied Sciences[26]	Case Study	Reproducibilty	Lack of reproducibility focused categories
Global Biomedical Literature[27]	Bibliometric Analysis	Misconduct-driven withdrawals in developing regions	Absence of a specific taxonomy
Retraction Watch[28]	Statistical Analysis	Data related issues dominate with rise in multi-cause cases	Lack of standardized taxonomy
COVID-19 Publications[29]	Comparative Analysis	Increased withdrawal due to rapid publishing	Temporal Bias
Multicenter Clinical Trials[30]	Meta Analysis	Protocol deviations and reporting inconsistencies	Insufficient process oriented classification
Nursing & Medical Education[14]	Systematic Review	Ethical violations reported 80% cases	Domain specific bias limits generalization
Multidisciplinary Datasets[31]	NLP-based machine learning	Multi label classification improves scalability	Dependent on notice quality
Cross-disciplinary Journals [32]	Mixed method Analysis	Distinction between error vs misconduct	Inconsistent reporting formats
Open-Access Journals[33]	Cross-Sectional Study	Higher visibility linked to Faster detection	Bias towards indexed/open access sources
AI- assisted scholarly datasets[34]	Classification	Improved Accuracy in detecting nuanced reasons	Limited model explainability

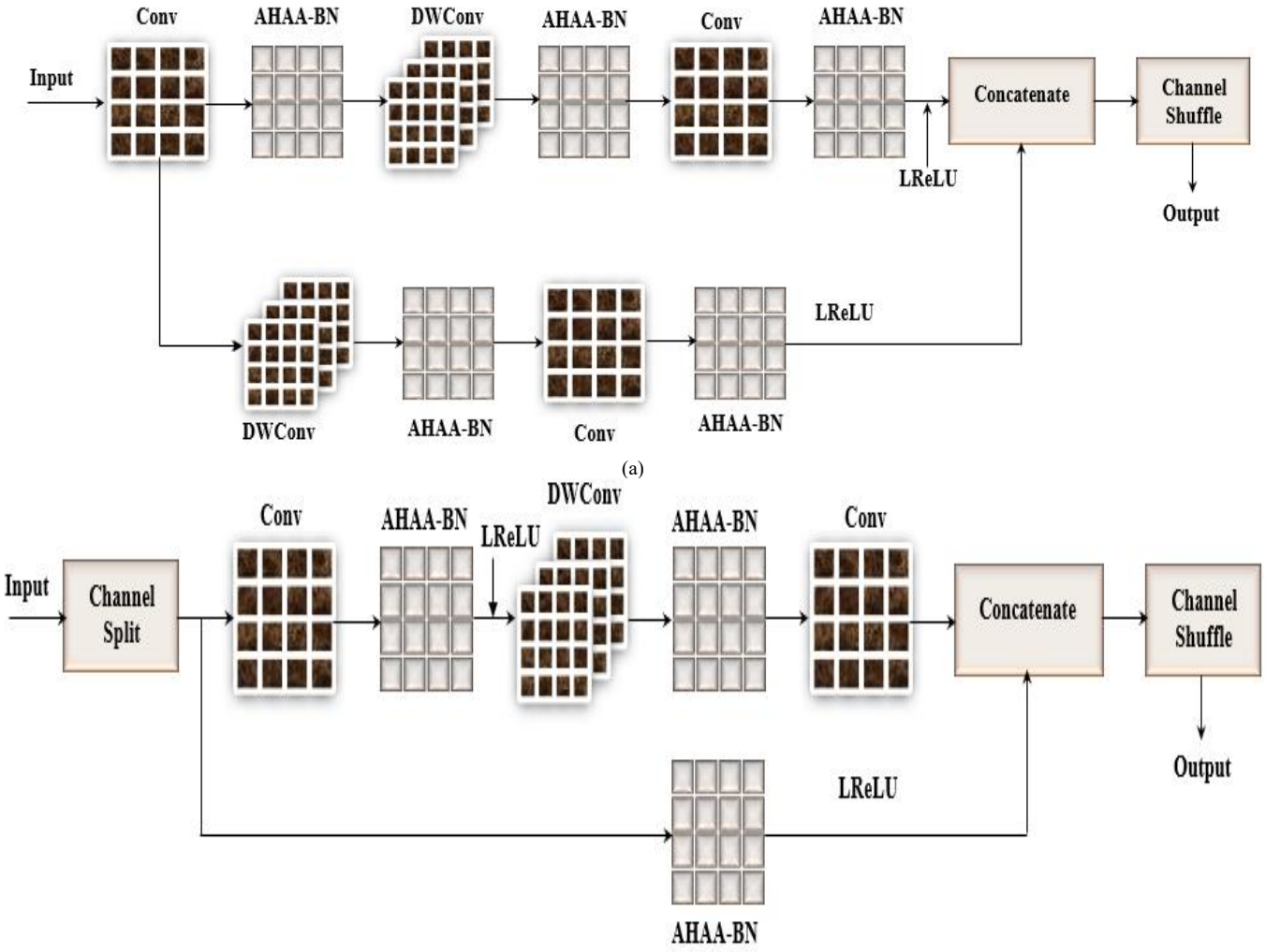


Fig. 4. Depiction of the modified Stage 3 in the AAHA-BN-ShNet model: (a) Non-layered structure; (b) Layered structure.

AttendEase: Fraud-Resistant Attendance Tracking in Higher Education Using Dynamic Tokens and Geofencing

Silvia Gaftandzhieva, Gabriela Lyavova
University of Plovdiv Paisii Hilendarski, Plovdiv, Bulgaria

***Abstract*—Preparing qualified specialists depends directly on student engagement during the educational process. For this reason, many higher education institutions (HEIs) view attendance as a key element of their educational policy and mandate strict requirements in internal regulatory documents. Traditional student attendance tracking is often time-consuming and susceptible to proxy-scanning fraud. Existing automated solutions, such as RFID or biometric systems, require expensive hardware infrastructure, while static QR-code approaches fail to prevent remote image sharing. This study proposes AttendEase, a cross-platform mobile application that secures and automates attendance management using a Bring Your Own Device (BYOD) paradigm. The system introduces a four-layer validation mechanism that combines time-limited dynamic QR codes (UUIDv4), duplicate registration, device-binding verification (SSAID/IDFV) to prevent multi-account logging from a single smartphone, and location validation using a Haversine-based geofencing algorithm. Beyond automated check-in, AttendEase features an analytics module that enables educators and students to generate real-time visualisations and reports with aggregated data. The empirical evaluation at Plovdiv University across 5 courses involving 48 students demonstrates the system's initial feasibility, usability, and potential to streamline attendance tracking, eliminate proxy check-ins, and reduce administrative overhead during lectures. The findings confirm that AttendEase has high potential to support effective attendance monitoring, operational management, and data-driven decision-making in HEIs.**

Keywords*—Attendance tracking; QR code; geofencing; flutter; cloud systems; mobile security; educational technology, smart campus

I. Introduction

Higher education plays a critical role in preparing qualified specialists and developing the knowledge and skills that students need to succeed in modern society. The quality of this process depends on various factors, among which the students' active participation holds a particularly important place. A key indicator to measure this participation and engagement is attendance at lectures, seminars, and laboratory sessions, as regular classroom presence directly affects learning outcomes and overall academic performance [1].

Attending lectures allows students to receive systematised information, practical examples, and teacher guidance that are difficult to fully replicate through self-study, ask questions, and receive immediate feedback. Its importance is especially critical during seminars and laboratory sessions for students in engineering, technical, and natural science programmes, where they apply theoretical knowledge in practice and develop skills in using specialised software, measurement systems, and equipment. Regular attendance also builds discipline, responsibility, and a professional attitude. Therefore, many HEIs consider attendance a vital component of the overall educational process [1] and require a minimum number of attendances for semester validation or exam admission. For instance, according to the Regulations of Plovdiv University Paisii Hilendarski [2], attendance at lectures, seminars, laboratory sessions, and practical exercises is mandatory and monitored by the respective instructors and department heads. This makes accurate and reliable attendance tracking critically important.

On the other hand, attendance information is a valuable tool for teachers and administrative staff. It indicates interest in specific subjects and teaching effectiveness, and can improve educational planning and compliance monitoring. Low attendance can signal organisational problems, insufficient motivation, an unsuitable schedule, or challenges in grasping the material. Therefore, collected attendance data can be used as feedback to improve educational quality.

Despite the key significance of this indicator, the process of registering and tracking attendance is still carried out conventionally in many HEIs - via paper sign-in sheets with manual signatures, oral roll calls, or subsequent manual entry of data into spreadsheets. These methods waste instructional time, increase the risk of technical errors, create difficulties with subsequent information processing, and limit analytical and reporting capabilities. There are also opportunities for abuse, such as marking a student who is actually absent as present.

For this reason, modern automated attendance tracking solutions are needed. With advancements in mobile technologies, cloud services, and wireless communication, new opportunities emerge to modernise attendance tracking through solutions based on QR codes, geofencing, online synchronisation, and rapid report generation.

To overcome the disadvantages of traditional methods, facilitate teachers' work, improve data accuracy, eliminate opportunities for fraud, and give students fast, transparent access to information, this article presents the mobile application AttendEase, which automates student attendance registration. The application allows students to register their presence by scanning a QR code dynamically generated by the

teacher. Before registering attendance, it checks the user's location against defined geographical coordinates, the scanned code's validity, duplicate registration, and the device's identification number used for registration. AttendEase supports user role management and automated, real-world, real-time attendance report generation. The data is stored in a cloud database, ensuring centralised management, reliability, and availability. The developed application is ready for deployment in a real university environment and can be easily extended with additional functionalities.

The study is organised as follows. Section II analyses existing attendance registration approaches. Section III describes the application's conceptual model. Section IV outlines the practical realisation of AttendEase's core functional modules. Section V presents and analyses the results of the application's experimental testing in a real environment. The conclusion outlines a plan for the system's future development.

II. Literature Review

As information technology has developed, various approaches and systems for automated attendance tracking have emerged, using a variety of student identification methods. A common approach to tracking attendance is using paper lists. In this method, the teacher records attendance manually, or students sign themselves on a shared sheet. This method requires no technical tools but consumes significant instructional time in large groups. During the subsequent processing, there is a continuous risk of technical errors, such as marking an absent student present. Another problem is the necessity for physical archiving and document management. As an alternative to paper lists, contact-free Radio Frequency Identification (RFID) access cards have emerged [3-5], in which each student scans their smart card on a specialised reader at the classroom entrance. The database stores the captured information, which allows subsequent processing and analysis. This approach automates registration and saves time, but requires capital investments to purchase and install specialised hardware equipment (RFID readers) and build infrastructure to connect to the information system. The main disadvantage is weak fraud protection, as a student can easily scan absent peers' cards. Biometric systems can also be deployed to register attendance, recognising unique biological characteristics such as fingerprints, facial features, palm geometry, or the iris [6-7]. These systems provide the highest degree of identity validation, practically eliminate proxy registration, and, when configured correctly, offer fast identification. The main disadvantages include high equipment costs, bottlenecks at classroom entrances for large student flows, and strict legal restrictions on storing sensitive personal data under GDPR requirements. Two-dimensional barcodes that can be read with a mobile device's camera, Quick Response (QR) codes, are also used for rapid identification. In this approach, the teacher generates a QR code for a specific class session containing identifying details (e.g., subject, date, time) [8-10]. Students confirm their attendance by scanning the code, after which the data is sent to a server or cloud database for storage. The approach operates entirely on the Bring Your Own Device (BYOD) principle. It does not require the university to purchase specialised hardware, making it an economically viable solution with easy integration capabilities. A disadvantage of this approach is the lack of protection mechanisms against remote abuse, such as scanning the codes outside the lecture hall.

Based on the reviewed technologies, automated attendance registration solutions for educational institutions have been developed. Kassem et al. [3] develop a classical attendance registration system using personal RFID cards, and Kurniali and Mayliana [11] integrate RFID readers with a web-based information system for data analysis and report generation. Despite automation, both solutions remain vulnerable to card passing and require hardware investments. Other researchers investigate the applicability of biometric systems. Osasuyi et al. [12] present a fingerprint-based solution that guarantees personal authentication, while Smitha et al. [6] design a facial recognition system using static cameras in the room. These systems achieve high security but require substantial computational resources and data security enforcement when processing sensitive personal data. The greatest flexibility is demonstrated by solutions that require mobile device usage. Masalha and Hirzallah [13] prove that scanning a unique QR code for each class drastically reduces check-in times and facilitates information processing for the teacher. The main disadvantage of early QR systems was the possibility of abuse, such as photographing the code and sending it to a student outside the hall. To overcome this disadvantage, Taju et al. [14] and Gangurde et al. [15] propose solutions that combine QR with geolocation technologies to track device location in real time. Muthulakshmi et al. [16] utilise the concept of Geo-Fencing, where a virtual geographical zone is defined around the classroom. This significantly reduces the risk of unlawful attendance registration outside the defined perimeter.

The review indicates that while mobile applications and QR codes are the most practical and cost-effective solutions, they require additional protection mechanisms during implementation. In most existing systems, security against remote abuse (such as using another person's account or scanning outside the hall) remains unsatisfactory. Since local storage or semi-automated processing hinders subsequent analysis and timely report generation, data management presents yet another challenge. Last but not least, the complex user interface of some solutions hinders daily adoption by students and teachers. For these reasons, systems under development must ensure high reliability, ergonomics, and rapid deployment. To achieve this objective, we chose a hybrid approach for the AttendEase mobile application design and development, combining dynamic QR codes with automated checks for code expiration, duplicate registration, hardware identification numbers, and the device's current GPS coordinates relative to the hall's geographical perimeter. A cloud architecture is utilised for data storage, enabling centralised management and rapid real-time report generation. The developed application achieves high reliability, security, and efficiency without requiring HEIs to invest in hardware.

III. Conceptual Model

AttendEase application aims to replace traditional paper-based and manual attendance registration methods in HEIs through mobile technologies, QR codes, and a secure cloud

infrastructure. The functional and non-functional requirements of the system were defined following a detailed analysis of the needs of the primary user groups – teachers and students. Functional requirements determine the core capabilities provided to users, such as user authentication (login) into the system, user role management, schedule and class session management, dynamic generation of unique QR codes for specific sessions, QR code scanning by students, automated attendance registration and validation, ability for manual entry or correction of attendance by a teacher, centralised storage of session and attendance information, visualisation of statistical data and report generation. Non-functional requirements describe the system's quality characteristics, such as secure encryption and storage of user data, protection against unauthorised access and tracking fraud, high processing speed and minimal request processing time, a convenient, responsive, and intuitive user interface, seamless operation across a wide range of mobile devices, and efficient cloud database connectivity and real-time synchronisation.

The workflow follows a strict logical sequence. After successfully logging in, the user is redirected to an interface personalised according to their role. The teacher activates a new session for the current class, and the system displays a unique QR code (e.g., on a multimedia screen in the room). To allow organisational flexibility, the system allows teachers to start sessions and generate QR codes up to 7 days in advance. This allows teachers to pre-print codes on paper or embed them in presentation materials. Students authenticate their attendance by scanning the code through the mobile application. After the system executes the embedded verification algorithm, it records the attendance in the database. This approach increases the process' speed and reliability and reduces errors. If issues occur, the teacher can make manual corrections (adding or removing attendance records).

QR code usage provides a quick and convenient mechanism for session registration. Without additional verification, the QR code could be shared with absent students, or students could hand over their accounts to peers to check in on their behalf. To ensure reliability and eliminate fraud during attendance tracking, AttendEase applies multi-factor verification to prevent check-ins during expired sessions, lateness, duplicate code scanning, remote scanning, or use of someone else's device.

Let S is the set of students, R the set of classrooms, L the set of all class sessions, Q the set of all initiated QR sessions, TC the set of instructors, and T a continuous time space. An ordered 11-tuple describes each active QR session $q_k \in Q$:

$$q_k = < id_k, l_j,\ teach_n, t_{start},\ t_{expire}, StatusQR_k,\ Hash_k,\ (locLat, locLng),\ maxDistMet, sessionDat >, \quad (1)$$

where, $StatusQR_k \in \{Active, Expired, Closed\}, teach_n \in TC$ is the identification number of the teacher who created the session, $t_{start} \in T$ and $t_{expire} \in T$ are the defined start and end of the session (code validity), $Hash_k = H(id_k, l_j, t_{expire})$ is a cryptographic token, $(locLat, locLng)$ are the coordinates of the classroom's centre, $maxDistMet > 0$ is the permitted deviation radius in meters, and $sessionDate$ is the session date.

An attendance registration request by a student $s_i \in S$ for a given lecture l_k in room $r \in R$is represented as:

$$Req = < s_i, Hash_k, q_k, t_{req}, (lat_s, lon_s), DevID_s >, \quad (2)$$

where, $Hash_k$ is the transmitted token from the scanned QR code, $t_{req} \in T$ is the system time when the request is sent, (lat_s, lon_s) are the student's GPS coordinates at the moment of scanning, and $DevID_s$ is the device's hardware identifier (SSAID/IDFV).

The validation decision for the attendance registration $V(Req) \in \{Approved, Late, Pending\_Approval, Rejected\}$ is determined after validating the dynamic token's status and time interval, record uniqueness, location, and the device identifier.

The dynamic QR code is generated $u_s = UUIDv4()$ with generation time t_{gen}. The request is valid if $\Delta t_{token} = t_{req} - t_{gen} \leq \Delta T_{Max}$, where ΔT_{Max} is the allowed grace period delay. The time and validity status predicate $P_{time}(Req)$ verifies if the session is marked as active and if the time window has not expired:

$$P_{time}(Req) = (statusQR_k = active) \cap (t_{req} \leq t_{expire}) \quad (3)$$

The lateness status relative to the class start t_{start} is determined as:

$$Status_{time} = \begin{cases} Approved, if\ t_{req} - t_{start} \leq 30\ min \\ Late, if\ t_{req} - t_{start} > 30\ min \end{cases} \quad (4)$$

Let $A \in S\ x\ L\ x\ Q$ be the set of all successfully confirmed attendance records, and $A = \{< s_i, l_j, q_k, t_{req} >\}$ is an already recorded attendance for student s_i for session q_k. A duplicate entry request will be blocked if such a record already exists for the current session q_k:

$$P_{uniq_{rec}}(Req) = \neg \exists a A: (a.s = s_i \cap a.l = q_k.l_j) \quad (5)$$

The distance $d(s,r)$ between the student (lat_s, lon_s) and the coordinates of the room's centre $(locLat_r, locLon_r)$ is calculated using the Haversine formula:

$$\Delta lat = \frac{\pi}{180} * (locLat_r - lat_s)$$

$$\Delta lon = \frac{\pi}{180} * (locLon_r - lon_s) \quad (6)$$

$$\alpha = sin^2\left(\frac{\Delta lat}{2}\right) + cos\left(\frac{\pi}{180} * lat_s\right) * cos\left(\frac{\pi}{180} * locLat_r\right) * sin^2\left(\frac{\Delta lon}{2}\right)$$

$$c = 2 * atan2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d(s,r) = R_E * c\ , where\ R_E = 6\,371000\ m$$

The location validity condition is $d(s,r) \leq R_{allowed}$, where, $R_{allowed} \leq 50$ meters.

Let $DevID_{registred}(s)$ be the pre-registered unique device ID of the student s in the database. The check will return:

$$Match_{device} = \begin{cases} True, if\ DevID_s = DevID_{registred}(s) \\ False, if\ DevID_s \neq DevID_{registred}(s) \end{cases} \quad (7)$$

AttendEase is designed as a client-server system. The mobile application serves as the client-side interface through which users interact with the system. The server side is based on cloud infrastructure that provides Backend-as-a-Service (BaaS) managed components for authentication, distributed data storage, real-time synchronisation, and user session management. To achieve a loosely coupled architecture and high maintainability, AttendEase is divided into 4 core functional modules:

- Authentication Module – manages user identification, access rights verification, and automatic redirection to the home screen based on the defined role.
- Session Management and QR Code Generation Module – maintains information regarding subjects, classes, teachers, student groups, and semesters, and manages the code generation process.
- Attendance Registration and Validation Module – enables scanning and processing of QR codes, executing automated checks for code validity, preventing code reuse, remote proxy check-ins (comparing the device's current coordinates with the classroom's designated location), and account handovers (checking if the registration device matches the pre-registered one).
- Analytical Module – provides instructors and students with structured access to data on conducted sessions, attendance counts, absences, lateness, and percentage activity, with information fetched from the database in real time and visualised via lists, charts, and/or aggregated metrics.

The AttendEase architecture is designed for scalability. Cloud service usage facilitates integration with university information systems and serving a large number of users without significant changes to the application structure.

IV. Software Realisation

Designing a modern multi-platform application requires selecting software technologies that guarantee high performance and scalability. This section presents the system's structure, interface solutions, user roles, database organisation, and the realisation of core functional modules.

A. Development Technologies

AttendEase is implemented using technologies for creating a cross-platform solution for Android and iOS. The Flutter framework was chosen as the primary development environment, which ensures the construction of multi-platform applications from a single codebase with a modern appearance and an excellent user experience. Dart was selected as the programming language because of its high performance and convenient tools for user interface design. A major advantage of Flutter is the ability to develop apps for Android and iOS simultaneously, which enhances applicability, reduces development costs, and simplifies maintenance. Additionally, the framework provides ready-made integration tools for cloud services. It has a rich ecosystem of external libraries (including QR code processing, coordinate extraction, permission checks, and distance calculations), an active user community, and exhaustive documentation.

For data storage and real-time synchronisation, the Google Firebase platform was used, including specialised services for authentication, data storage, and push notifications.

The document-oriented NoSQL database Cloud Firestore was selected as the database management system. The main argument for this choice is its native support for real-time automatic synchronisation. When a QR code is scanned and attendance is recorded, the application interface updates automatically, enabling attendance monitoring during an active QR session. Data protection is secured at the cloud level through Firestore Security Rules, which define read and write permissions according to the user's role.

The system's core business logic is realised through dynamic QR code generation and scanning technologies, alongside location tracking services via mobile devices' built-in GPS modules. Combining these two technologies eliminates opportunities for fraud involving capturing and forwarding the code image to absent students. The main reasons for choosing QR technology are its flexibility in encoding heterogeneous content (text, numerical values, identification tokens, and coordinates), the presence of positioning markers, and a built-in Reed-Solomon error correction algorithm. This allows reliable decoding even under partial smudging, poor angles, or low projector resolution in the lecture hall. Process security is enhanced by deploying dynamic QR codes with narrow time windows and automatic disabling after a specified number of minutes. As an additional security mechanism, AttendEase utilises geolocation via the devices' embedded Fused Location Services, combining data from GPS, Wi-Fi networks, cellular towers, and other sensors for accurate positioning. The system calculates the distance between the student's position and the pre-configured classroom coordinates using the Haversine formula. If the calculated distance exceeds the defined threshold (e.g., 50 meters), the system rejects the transaction. To comply with GDPR, the application processes location data solely in the device's local memory during scanning, ensuring no geographic coordinates are permanently stored in the cloud.

The development and testing toolkit includes the Visual Studio Code integrated development environment (IDE), Flutter SDK, Firebase Console for cloud infrastructure management, external libraries, Git and GitHub (for decentralised version control, change tracking, and archive repository management), software emulators, and physical Android and iOS mobile devices for testing AttendEase functionalities.

B. Project Structure

The application utilises a mixed architectural model that combines layer separation with grouping user screens per role. At a conceptual level, the system is split into three main layers:

- Presentation Layer – encompasses the user interfaces (UI screens) implemented in the `lib/screens` directory and visual components in the `lib/widgets/` directory.

- Business Logic Layer (Service Layer) – encapsulates the application's business logic (`lib/services/` directory) and contains classes for communicating with the REST API and the Firebase cloud platform, session management, geolocation handling, and system notifications.
- Data Layer – defines the structure of core objects in the system (`lib/models/` directory) - users, subjects, classes, attendances, and QR sessions.

In the root `lib/` directory, the `main.dart` file serves as the entry point for the software lifecycle, initialising cloud services and global states. Helper components (validators, extended date and time formatters) and visual themes are distributed across the `utils/`, `theme/`, and `l10n/` subdirectories (for interface internationalisation and localisation).

Fig. 1 presents the AttendEase architecture.

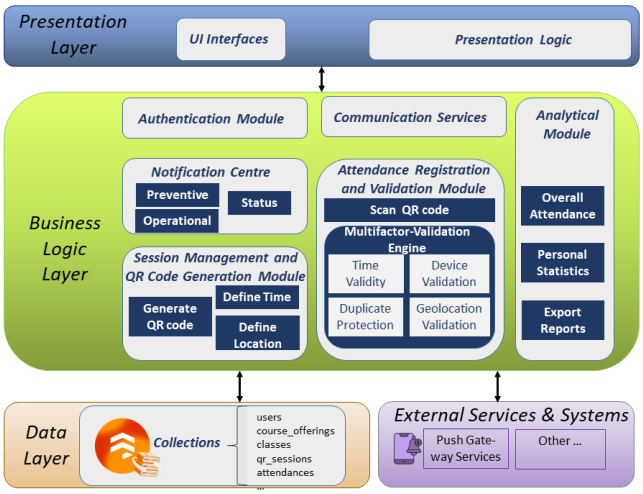


Fig. 1. Architecture.

AttendEase realises state management through a hybrid approach. The Provider architectural pattern, combined with ChangeNotifier, is used for data that affects the entire application (authentication status, current user profile, and role). Ephemeral states of individual screens (loading indicators, selected tab, QR scanner activity) are managed via the standard setState mechanism, optimising performance and resource consumption. The architecture minimises coupling between layers - components in the presentation layer do not communicate directly with the database. Communication is conducted by invoking methods from the service layer. For single transactions, asynchronous requests (Future/Await) are utilised, while real-time changes are tracked using data streams through the StreamBuilder component and Firestore snapshots. AttendEase manages navigation between screens by declarative access control mechanisms that prevent unauthorised student access to administrative panels.

Although the project does not implement Clean Architecture in full, the chosen mixed model offers an optimal balance of readability, development speed, and maintainability. The main advantages of the proposed architecture are the clear separation between UI, business logic, and data, and the isolation of Firebase communications into independent services, which allows easy future migration to another backend infrastructure or local university information systems without modifications to the presentation layer.

C. User Roles and Access Model

The AttendEase architecture utilises a role-based access control model. This approach enhances personal data security, optimises the UI, and eliminates the risk of unauthorised actions. After successful authentication, the system loads the user token from Cloud Firestore and dynamically configures the accessible functionality.

Fig. 2 presents a Use Case diagram of the AttendEase mobile application.

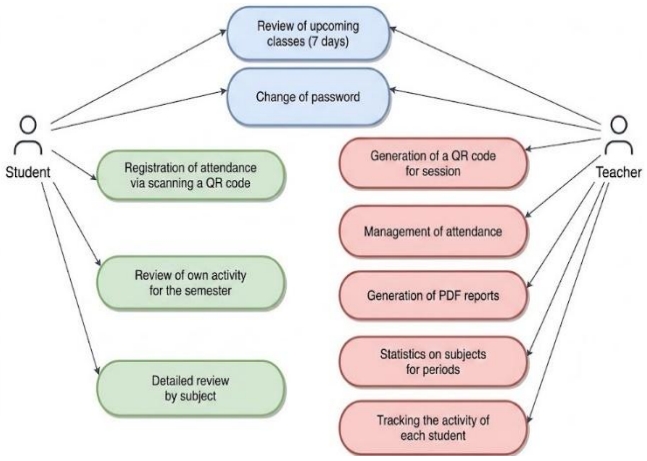


Fig. 2. Use case diagram of AttendEase.

The Student user role is designed for learners whose interaction with the system is restricted to registering attendance and reviewing their own attendance history. Upon logging into the system, the student has access to:

- Home screen – displays an academic schedule with planned sessions for 7 days;
- Attendance registration module – provides quick access to the mobile device's camera for scanning generated dynamic codes and automated feedback regarding request outcomes (successfully recorded attendance or rejection categorised by reasons: expired session, invalid code, lack of location permission, etc.);
- Analytical module – provides access to data on registered attendances and percentage ratios across subjects.

The Teacher user role is intended for individuals who conduct class sessions and manage attendance tracking. Upon logging into the system, the instructor receives access to an interface for:

- QR session management – enables selecting a specific class and generating a unique QR code with a validity period and location coordinates corresponding to the hall where the session takes place;

- Real-time student status monitoring (present, late, absent);
- Manual status correction during extraordinary situations or technical issues with a student's device;
- Detailed attendance statistics review at the student, group, or cohort level with capabilities to export into formats suitable for sharing or archiving (PDF/CSV).

D. Database

The database stores the information in logical collections. Table I describes the core collections, their attributes, and functional purposes.

TABLE I. STRUCTURE OF CORE COLLECTIONS IN THE DATABASE

Collection	Primary Fields (Attributes)	Functional Purpose
users	id, role, email, firstName, lastName	Stores information about system users
faculties	id, name, code	Stores information about faculties
specialties	id, name, code, degree	Stores information about study programmes
cohorts	id, specialtyId, startYear, endYear	Stores information about student streams/cohorts
groups	id, name, specialtyId, cohortId	Stores information about student groups
semesters	id, name, term, academicYear	Stores information about academic semesters
subjects	id, name, code	Stores information about subjects
course_offerings	id, subjectId, semesterId, cohortIds	Links subjects to a specific semester and student stream
classes	id, offeringId, teacherId, room, dayOfWeek	Stores information about individual class sessions
qr_sessions	id, classId, teacherId, token, expiresAt	Stores active QR sessions for attendance registration
attendances	id, classId, studentId, status, scannedAt	Stores information about registered attendances

Although Cloud Firestore is a NoSQL database, the data model implements relational logic through document references (Document IDs). This prevents data duplication across different locations and maintains connection integrity. To improve performance when processing large volumes of data, indexes have been configured to facilitate query execution, such as filtering attendances by a specific subject for a stream of students. The system ensures data-level security through declarative Firestore security rules. Students have read permissions only for their own attendance records, and only users with a teacher role can modify and write to qr_sessions and attendances. Thanks to built-in WebSocket support, the system monitors collection changes in real time via client interfaces and reactive components (StreamBuilder). If necessary, the designed database can be expanded, for instance, by adding new collections and attributes to store notifications and archived reports, or for integration with university systems.

E. Authentication Module

The authentication module provides reliable user identity validation. It implements the authentication process by integrating Firebase Authentication, eliminating the need for local password hashing and storage. Upon launching, the application checks for an active user session. If the token is active, the system redirects the user to the corresponding home screen. Otherwise, a screen with a login form appears, requiring the user to enter an email address and password.

Code Snippet 1 shows the logical sequence for authentication and user status verification. After successful matching of the entered credentials in the Firebase Authentication cloud directory, the system extracts the subject's unique identifier (`uid`). Then, it requests the `users` collection in Cloud Firestore to validate the profile status. If the document does not exist or the `isActive` attribute is false, `signOut`() terminates the session and sends an exception to the presentation layer. Upon successful verification, `notifyListeners`() updates the global Provider state. After successful authentication, the system loads the corresponding home screen depending on the user's role.

Code Snippet 1. User authentication via Firebase Authentication

```
final userCredential = await_auth.signInWithEmail
          AndPassword(email:email,password: password,);
final uid = userCredential.user!.uid;
final user = await getUserById(uid);
if (user == null) {await _auth.signOut();
      throw Exception('User not found in database.');}
if (!user.isActive) {await _auth.signOut();
        throw Exception('Account is deactivated.');}
_sessionUser = user;
_ready = true;
notifyListeners();
```

The application uses centralised session management via Firebase Authentication, enabling automatic session termination upon logout, protecting sensitive information, and ensuring secure credential storage. As an additional security measure, upon initial login with a system-generated temporary password, the system requires the user to change the password before granting access to functionalities. The modular structure permits seamless integration of additional security methods, such as password recovery mechanisms, multi-factor authentication, and biometric recognition via device hardware modules.

F. Session Management and QR Code Generation Module

Initialising an attendance session is performed by users with a Teacher role in 3 steps: reviewing the academic schedule and selecting a specific class session, defining geographical coordinates for conducting the session by using the mobile device's current location via the built-in GPS module or selecting classroom coordinates from an interactive map, and generating and visualising a QR code.

The system uses indirect encoding to enhance security. Instead of embedding explicit metadata about the subjects, study programme, or lecturer in the QR code, which could be easily decoded and shared, the system encodes a cryptographically secure, compact identification token in the matrix that serves as a secure key (reference) to the corresponding document in the cloud database. This minimises the risk of content counterfeiting or local manipulation of client requests. Code Snippet 2 shows how the application creates a new session. To generate a unique token for a class session,

UUIDv4 is used, and the result is parsed into a compact string to optimise optical recognition. The `QRSession` class object stores the class identifier (`classId`), teacher identifier (`teacherId`), creation and expiration timestamps (`createdAt, expiresAt`), current status, geographical coordinates (`locLat, locLng`), and the allowed radius fixed at 50 meters (`maxDistMet`). The data is written into the `qr_sessions` collection.

Code Snippet 2. QR session initialisation

```
final token = const Uuid().v4().split('-').first;
final session = QRSession( id: const Uuid().v4(), classId:
        classItem.id,teacherId: teacherId,createdAt:
         now,expiresAt:expiresAt, status: QRSessionStatus.active, token:
token, locLat:
    locLat, locLng:locLng, maxDistMet:50,
    sessionDate: formatSessionDate(sessionDate),);
await fs.createQRSession(session);
```

The QR code is displayed on the teacher's screen in high resolution. The interface provides additional session and location metadata, and options to export or download the graphic file (e.g., for visualisation on an external screen, embedding in presentation materials, or external systems). Each generated code has a strictly defined lifecycle, bound to the duration of the academic class hour.

The embedded QR code generation module in AttendEase provides a high-performance, cost-effective, and secure mechanism to identify academic sessions. Its software architecture allows easy future scaling and full integration with external university information systems.

G. Attendance Registration and Validation Module

Students can scan a QR code in the application. After logging in, the student can access the scanning function via a dedicated button on the home screen. After the student activates this function, the application opens the mobile device's camera and initiates a real-time QR code recognition algorithm. Code Snippet 3 shows a fragment of the QR code recognition implementation. After successful scanning, the code value is extracted, representing an active QR session token. The token is then returned to the main screen for subsequent validation and attendance logging.

Code Snippet 3. Scanning a QR code via the device camera

```
MobileScanner(controller: _controller, onDetect: (capture){
  if (_isHandlingDetection) return;
  final barcode = capture.barcodes.isNotEmpty?
          capture.barcodes.first: null;
  final value = barcode?.rawValue;
  if (value == null || value.isEmpty) return;
  _isHandlingDetection = true;
Navigator.pop(context, value);},)
```

The camera processes and displays the image in real time, showing a targeting frame or indicator for proper QR code positioning. After successful recognition, the application extracts the token and sends a request to the database to check the corresponding active QR session. Before recording attendance as a new document in the `attendances` collection, the system performs four checks to guarantee tracking reliability:

- Time validity – verifies whether the QR session exists and has an Active status and whether the current system time is less than `expiresAt`;
- Duplicate protection – verifies if an entry already exists for the same `studentId` and `classId` on the same date;
- Geolocation validation – checks whether the scanning device is within range of the location specified during the QR code creation by calculating the distance to the reference point;
- Device validation – checks for a match between the hardware identifier and that of the student's initially registered device.

The system applies an automatic logical constraint: if a student scans the code more than 30 minutes after the official start of the class, the transaction completes successfully, but the attendance status attribute in the `attendances` collection is logged as `Late`. After the class ends or after a specified time window, the session automatically transitions to `inactive` status, preventing late tracking attempts.

When the system receives a scanned token from a client device, it uses Firestore Security Rules and the corresponding service classes to verify whether the associated session in `qr_sessions` is `active` and whether the current system time does not exceed `expiresAt`. At the same time, it executes a verification query to prevent duplicate records for the same student in the same class.

Verifying users' locations via geolocation restricts off-site attendance registration and prevents non-present students from fraudulently logging in. The location verification procedure utilises Fused Location Services, which pull data from GPS modules, cellular networks, and Wi-Fi access points. When scanning is initiated, the client application requests permission to access geolocation services. Upon receiving permission, the system extracts the mobile device's current geographical coordinates (latitude and longitude) and calculates the distance `d` to the defined reference point using the Haversine formula. The `Geolocator`.distanceBetween method accepts the student's current (`lat, lng`) and the reference coordinates (`locLat, locLng`) from the current session as parameters. When the `distance` variable value exceeds the pre-configured limit in the `session.maxDistanceMeters`, the system generates a location validation exception and changes the attendance status to `Rejected`. In the current implementation, the system enforces a 50-meter geofencing threshold around the lecture hall. This value is good for indoor university spaces, as it allows some tolerance for GPS hardware deviations in buildings with massive reinforced concrete structures while preserving verification reliability.

Code Snippet 4. Calculating distance between current position and class location:

```
double _distanceToSession(QRSession session, double lat, double lng)
```

```
{return Geolocator.distanceBetween(lat, lng,
          session.locLat, session.locLng, );}
    final distance = _distanceToSession(session, studentPos.latitude,
studentPos.longitude);
    if (distance > session.maxDistanceMeters)
        { // attendance is rejected}
```

The application design provides facilities to handle exceptions. If location access is denied, the application prompts the user to enable the permission. Upon a second refusal, it automatically redirects the user to the device settings. If the system can't determine precise coordinates due to a lack of GPS signal, indoor environments, or technical limitations, it displays an error message, and it does not complete registration until it retrieves valid data. To ensure information security and personal data protection (in compliance with GDPR directives), the application fetches and uses the user's location only within the millisecond timeframe in which it processes the scanned token, and does not perform continuous background tracking.

Verifying the device used for registration enhances the reliability of the automated tracking process by limiting the voluntary handover of user accounts. With standard email and password authentication systems, an absent student can give their login credentials to a colleague, who can then log fictitious attendance on their behalf. To mitigate this risk, AttendEase implements an additional check via the Device Identifier (Device ID). The procedure for binding a student's profile to the device used is initiated automatically during the first successful authentication in the application. The system extracts the device's unique identifier and writes it to the corresponding document in the `users` collection in Cloud Firestore. Code Snippet 5 presents how the application extracts the device identifier depending on the operating system. For Android, it fetches the Android ID (SSAID), which is unique for each combination of application, device, and user profile. For iOS, it uses identifierForVendor (IDFV) to ensure a constant value for apps from the same hardware vendor. During each subsequent request to scan a QR code and record attendance, the application extracts the current device identifier and compares it with the stored value. If the two values match, the process continues normally. If the system detects a mismatch, it updates the record state in the `attendances` collection to "Pending Approval". Then it displays a message on the student's mobile device informing them that manual verification is needed.

Code Snippet 5. Fetching device identifier

```
    Future<String?> getDeviceId() async {try {
      if (Platform.isAndroid) {
        AndroidDeviceInfo androidInfo=await _deviceInfo.
                androidInfo; return androidInfo.id;}
      else if (Platform.isIOS) {IosDeviceInfo iosInfo = await
       _deviceInfo.iosInfo; return iosInfo.identifierForVendor;}
        return null;
} catch (e) {return null;}}
```

When the student uses a different device, the system does not block registration completely. The teacher's interface captures changes in real time via a reactive Firestore Stream and displays a notification about a registration attempt from a different device. It provides lecturers with detailed information, allowing them to manually confirm or cancel the attendance after a physical verification in the auditorium. Upon confirming attendance, the system offers the option to update the registered device ID number. This capability to change devices is particularly useful if a student has replaced their smartphone, is temporarily using another device, or has reinstalled the application.

To ensure compliance with GDPR directives, the module does not collect or process sensitive personal data (such as IMEI numbers, phone numbers, or biometrics) and stores only anonymised strings. Combining device validation with QR codes and geolocation checks creates a multi-layered defence mechanism. The QR code validates participation in a specific class session, the Device ID check confirms the use of the registered hardware, and the location check proves physical presence within the designated zone. After the checks pass, the system displays a confirmation of successful registration on the client device, or a corresponding descriptive message in the event of failure or delay. Due to cloud data synchronisation within a few hundred milliseconds, the teacher can monitor attendance on their device screen in real time.

H. Analytical Module

Accumulated attendance data has real value only when it can be analysed and used for decision-making. Deploying an analytical module provides numerous practical benefits - teachers gain fast access to summarised information on student attendance, students have transparent access to their participation records, and academic leadership can use accumulated data to analyse activity patterns. The application sends a request to retrieve attendance information by student, subject, period, or group, performs statistical calculations, and visualises the summarised data as numerical indicators and via graphics to facilitate analysis. After a new attendance registration, data updates dynamically without requiring a manual screen reload.

Teachers can access data for all subjects they teach, select a specific one, and receive attendance data for all enrolled students. The system allows filtering by time periods (week, month, semester) and can visualise summarised data for attendance statuses, the attendance ratio for the selected period in a pie chart, and individual student attendance percentages for the chosen period in a bar chart. The module allows the teacher to export a report for each subject containing summarised attendance information for each student throughout the semester, to exercise precise control over participation in the educational process and semester validation.

Students can review their attendance and absence data over the semester, along with summarised attendance and absence figures across the subjects, enabling them to take timely measures to increase attendance to secure semester validation at the end of the term. Beyond aggregate values, the student can find detailed information for individual classes. For each session, AttendEase displays the date and time, subject, and attendance status (e.g., present, late, absent, or rejected due to location).

Thus, the embedded reporting and statistics module transforms AttendEase from a pure attendance logging system into a comprehensive tool for analysing and managing educational activity. Through clear visualisation and fast access to information, the system successfully supports and optimises activities for both instructors and students.

I. Interface and User Experience

The design and implementation of user interfaces (UI/UX) in AttendEase adhere to modern standards of ergonomics, intuitiveness, and responsiveness across varying mobile device screen dimensions. Architectural separation of the presentation layer ensures that visual components remain independent of business logic while providing reliable real-time user feedback.

The primary objective in creating the interface was to achieve maximum clarity and provide rapid access to core functionalities. Although the application is developed for a live educational environment, students must be able to scan the generated QR code immediately, and instructors must be able to initiate a lecture session or review/edit attendance with minimal interactions. The interface is built with Flutter and strictly adheres to modern Material Design specifications. Modern visual components, cards for logical data grouping, rounded buttons, smooth screen transitions, and a consistent colour scheme are utilised. The highly legible font used ensures good visibility across various screen resolutions. Font sizes and element spacing were selected to make information easy to consume without visually overloading the user. To adapt to external lighting conditions and user preferences, the application supports two themes: a light theme that provides high contrast and optimal legibility in daylight, and a dark theme that reduces ocular strain and fatigue in low-light environments. After logging in, the user is directed to the appropriate home screen. Students can register attendance by scanning QR codes, review the upcoming sessions schedule, and view personal attendance statistics. Teachers have access to an administrative panel that allows them to initialise and manage QR sessions, review attendance sheets, visualise attendance-monitoring charts, and export reports.

To minimise user errors and build trust in the system, it displays detailed context-specific error messages. After successful attendance logging, the student receives a confirmation. If an error occurs (e.g., invalid location or expired time window), AttendEase displays specific information about the problem.

In the user interface implementation, visual elements are leveraged for rapid information consumption - colour indicators, icons, progress bars, and graphs. For instance, low attendance is denoted with warning colours (red/orange), while high activity is highlighted in green.

From a technical standpoint, AttendEase uses the standard Flutter sizing mechanisms to ensure interface responsiveness. Elements reorganise flexibly using scrollable views, expanding containers, and adaptive layouts, ensuring proper visualisation across diverse mobile devices. The application is optimised for portrait mode operation. To enhance user experience, AttendEase minimises screen load times, section transitions, and core action processing times to ensure high system performance.

To improve user experience and facilitate educational organisation, a specialised notification and reminder subsystem was integrated into the application, optimising academic processes and ensuring a reliable, two-way informational stream between the backend infrastructure and end users. Through this subsystem, students and teachers can receive timely information regarding upcoming classes, important events, and potential issues relating to attendance logging. AttendEase manages push notifications through the Firebase Cloud Messaging (FCM) platform. Push notifications can notify users about critical events related to attendance tracking. During initial user authentication, the application requests permission to access the operating system's messaging subsystem and generates a unique FCM token for that specific client device.

Code Snippet 7. Syncing FCM token for push notifications

```
Future<void> syncTokenForUser(String userId) async {
  final settings = await _messaging.requestPermission( alert: true,  badge: true,  sound: true, );
  if (settings.authorizationStatus == AuthorizationStatus.denied) return;
  final token = await _messaging.getToken();
  if (token == null || token.isEmpty) return;
  await _saveTokenForUser(userId, token);
}
```

Code Snippet 7 illustrates how the application requests notification permissions and retrieves an FCM token for a specific device. The application saves the retrieved token in the database, enabling push notifications to be directed to the corresponding user. To ensure data security and prevent messages from being sent to unused hardware, logging out or switching devices automatically deactivates or updates the token. For automated monitoring via Firebase Cloud Functions, a cloud function was developed to track database transactions in real time. After it detects an attendance registration attempt outside the pre-configured radius, the system can automatically dispatch a push notification to the instructor, allowing a timely response and physical inspection. Unlike push notifications, local reminders are scheduled and generated autonomously at the operating system level, without requiring continuous internet connectivity. These notifications alert students and teachers to upcoming classes and reduce the probability of missing sessions. Upon successfully loading the user profile, the application fetches the current upcoming class schedule and schedules corresponding local alarms. In the current software version, the notification subsystem initiates two consecutive events for each class session: a preventive notification generated 15 minutes before the fixed start time and an operational notification generated precisely at the start of the session (see Fig. 3). The algorithm embeds native support for devices' local time zones, guaranteeing precise alignment of notifications with the actual academic schedule regardless of the user's geographical location.

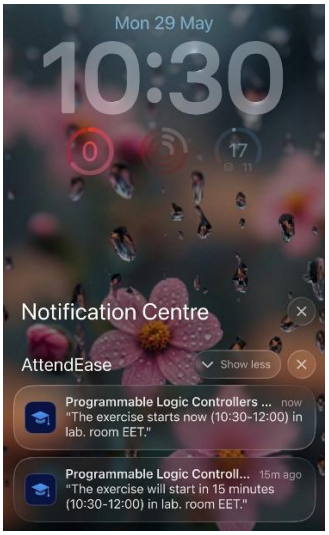


Fig. 3. Showing AttendEase alerts on lock screen.

V. Experimental Testing and Validation

To evaluate how effectively AttendEase meets teachers' and students' needs and operates reliably under real-world conditions, AttendEase has been tested at the University of Plovdiv Paisii Hilendarski. The experimental testing involved 1 teacher and 48 students utilising the application on their personal mobile devices under conditions simulating a live academic environment (immediately before and during class hours). Participants downloaded and installed the mobile app via the Google Play Store/Apple App Store and provided access permissions for the device camera and location tracking during operation. All participants received unique login credentials to access the system. To verify application compatibility and stability, testing was performed across physical iOS and Android devices, validating core operation scenarios: authentication processes for varying roles, including redirection to corresponding panels, enforced mandatory initial password changes, and accuracy of push notifications for upcoming classes (sent 15 minutes prior and at the exact start hour).

The testing process simulated real-world stages of logging class attendance - session preparation and launch, scanning, validation, and attendance data analysis. For experimental purposes, attendance tracking simulations were conducted for 5 subjects: Energy Management and Electrical Apparatus, Internet of Things, Programmable Logic Controllers, Optical Communication Technologies, and C# Programming Part 2.

Testing began with the teacher logging in to the application's home screen, which verified correct extraction of the current date, upcoming daily classes, and the planned schedule. Fig. 4 presents the process of initiating a QR session for "Programmable Logic Controllers". The teacher selected a specific class (10:30-12:00 on June 15) from the subjects list (Screen 1), set a geographic reference point using the location setting (EET Lab) and the map (Screen 2). After the teacher tapped the "Start QR" button, AttendEase generated a unique identifier and displayed a temporary QR code (Screen 3) on the device screen, valid for the duration of the class.

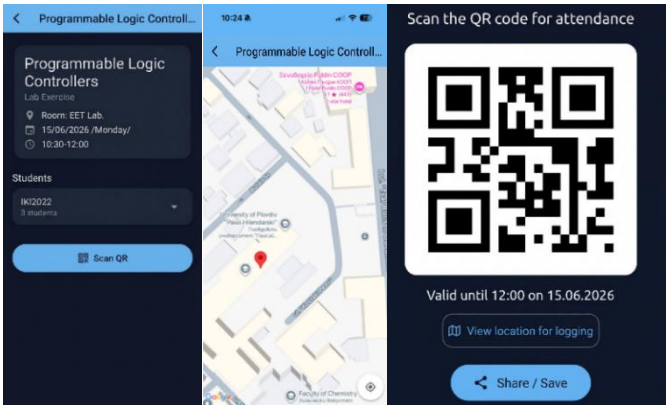


Fig. 4. Creating a QR session steps.

After the QR session activation, users with a Student role logged in to the application, reviewed the daily schedule, activated the scanning feature, and scanned the code displayed by the teacher using their mobile devices' cameras. The application recognised the code, extracted the required data to initiate registration, and executed automated background verifications to check session activity, time windows, uniqueness, geolocation constraints around the hall, and hardware ID matches against the initial profile bind. The system was tested against all forms of potential fraud. In every instance, the application executed the designated automated checks, responded according to embedded business logic rules, successfully identified attempts for registration abuse, marked records as "potentially suspicious" on the teacher's panel, and displayed appropriate alert notifications to students. AttendEase accurately determined and displayed student statuses.

During attendance registration, the teacher was monitoring incoming entries in real time on the tracking screen. All incoming check-ins were correctly registered as attended (highlighted in green), late (highlighted in orange), pending confirmation (highlighted in yellow), or no data for incomplete scanning processes (highlighted in grey), and then written to the cloud database (Fig. 5). To assess the system's fraud-detection capabilities in edge cases, we simulated a scenario where a physically present student attempted to log attendance using a peer's hardware. The teacher successfully performed a manual status override through the control panel and updated the device data (Fig. 5, Screens 3 and 4). While the anti-fraud testing evaluated basic location mismatches and device handover scenarios, this phase did not systematically benchmark explicit countermeasures against advanced location spoofing (such as Android mock-GPS tools or root-level location masking). Addressing hardware-level or application-level mock-location detection represents an essential vector for future security enhancements.

In the final stage, users tested the analytical module reporting capabilities. The teacher accessed analytical dashboards to review aggregate attendance data across subjects, track overall student attendance percentages, and identify learners with low activity levels.

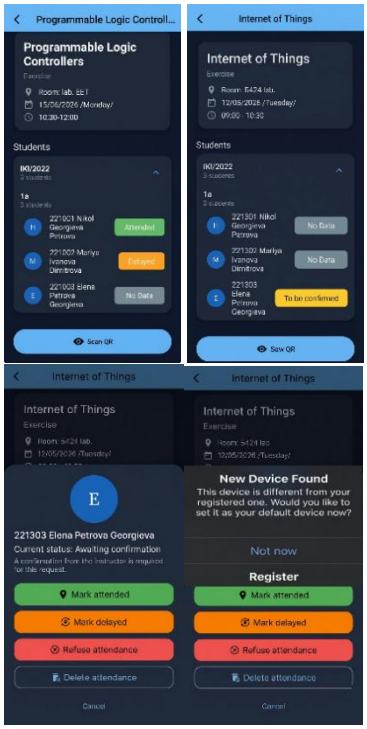


Fig. 5. Tracking the registration process details.

Fig. 6 presents visual dashboard screens with aggregate attendance data for the "Internet of Things" (Screens 1 and 2) and "Programmable Logic Controllers" (Screens 3 and 4) subjects over a selected period. For the chosen timeframe (18.05.2026 - 24.05.2026), 3 students were present, and 3 students were absent in the "Internet of Things" sessions. Aggregate semester data indicated that 3 students attended 67% of the conducted sessions, while the remaining 3 attended only 33%. During May 2026, 22 students (46% of the cohort) attended the "Programmable Logic Controllers" classes. Visualised data signalled that the teacher needs to deploy measures to increase engagement, particularly for students with attendance under 40% (highlighted in red). A comparative time trial showed that taking attendance for a group of 22 students in the "Programmable Logic Controllers" classes took an average of 3.5 minutes using traditional manual roll call. In contrast, AttendEase reduced this duration to 25 seconds in total, as students validate their presence concurrently. This yields an approximate 88% reduction in administrative overhead per seminar session.

A comprehensive summary of present logs (green), lateness (yellow), absences (grey), and rejected entries (red) is accessible for each subject (per student). Fig. 7 displays aggregate attendance data for the "Internet of Things" (Screen 1) and a detailed exported report (Screen 2). The data indicates, for example, that student number 221301 attended 3 sessions and was absent for 3 sessions.

Students tested the personal statistics section, tracking their attendance across enrolled subjects, total sessions conducted, successfully tracked attendances, and calculated attendance percentages for each course. In all cases, the system correctly computed attendance rates and systematised the information. Fig. 8 displays screens with aggregate summaries of attendances, absences, lateness, and rejected registrations for a student across all semester courses (Screen 1) and detailed historical status data for "Internet of Things" sessions (Screen 2).

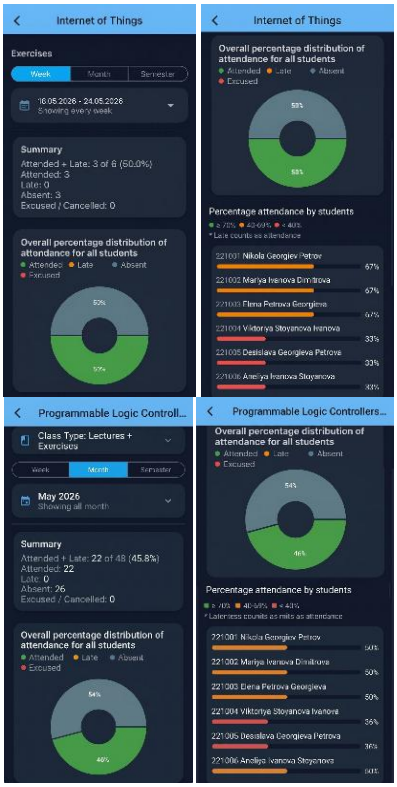


Fig. 6. Analytical panel visualisations.

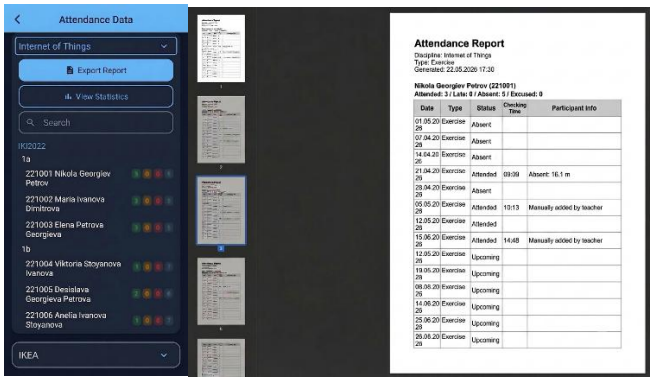


Fig. 7. Analytical panel**:** Aggregate data export example.

Tests proved that the application interface enables action execution in minimal steps, making it suitable for live classroom deployment where attendance tracking must be completed rapidly and without friction right before class starts. All automated verifications (time, geolocation, and Device ID) were triggered flawlessly, capturing and flagging all abuse attempts. Attendance data visualisation substantially facilitates analysis and identification of low-activity students. No critical errors compromising system functionality were identified during experimental testing. The minor interface inconsistencies identified were promptly resolved during iterative development.

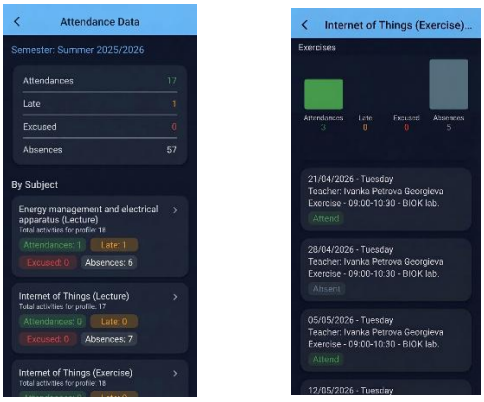


Fig. 8. Analytical panel: student views.

This experimental testing focuses on validating AttendEase's integrated four-layer verification end-to-end rather than comparative benchmarking. Empirical measurement of the incremental fraud-resistance provided specifically by the device-binding layer remains a topic for future comparative testing.

VI. Conclusion

The empirical observations from the prototype deployment align with findings in prior literature regarding the efficiency of QR-based attendance [13]. While earlier work cited check-in speed improvements, AttendEase prioritises a security balance, and the extra latency introduced by cloud-side multi-factor checks (time, boundary, device binding) trades off raw check-in speed for fully automated verification.

Functional testing across physical Android and iOS hardware confirms AttendEase's reliability and practical applicability as an effective automated student attendance tracking system in a real university environment. Its UI is engineered with a focus on ergonomics, operational clarity, and process efficiency. By applying modern design concepts, adaptive role-based navigation, and software optimisation, the application provides a high-quality user experience tailored to the academic community's specific needs.

AttendEase is a comprehensive mobile solution for automated student attendance tracking. Driven by its modular architecture, embedded verification mechanisms, and modern development technologies, the system offers a reliable, efficient instrument for optimising administrative processes in HEIs. AttendEase significantly shortens roll call time at the start of classes, eliminates tracking fraud and the risk of human error, ensures high data fidelity, facilitates daily lesson organisation, reduces administrative overhead for tracking and analysis, and fosters active student participation in the educational process.

Although the implemented mechanism for location verification and dynamic QR code scanning demonstrated a 100% success rate in simulated scenarios, its practical application in a real university environment presents certain technical limitations. The first one is related to geolocation and satellite signal drift (GPS drift). Receiving a GPS signal inside massive reinforced concrete buildings often causes interference, leading mobile devices to exceed the set 50-metre radius and potentially creating false negatives. In the current version, this issue is addressed by allowing teachers to register manually. To achieve full reliability, future versions of AttendEase plan to implement a hybrid geo-validation mechanism that combines GPS data with Bluetooth Low Energy (BLE) Beacons and checks for connectivity to the university network when the satellite signal is weak. This will ensure high precision and micro-location accuracy down to a specific desk in the lecture hall. The second challenge stems from scalability during peak loads and poor network connections. During lectures with a large number of participants (150–200 students), students will scan dynamic QR codes simultaneously within a 10–15 second timeframe. Sending dozens of requests to Google Cloud Firestore simultaneously can lead to temporary delays, request timeouts, or exceeding the per-second request limit. To address this issue, in addition to the currently implemented offline caching, plans include introducing a local queue for asynchronous data submission and using a message broker to handle peak traffic. Additionally, a future version will enable generating a temporary unique offline code based on the student's Device ID, which the lecturer can scan or enter manually at the end of class when connectivity is absent.

Another methodological limitation is the sample size: 1 teacher and 48 students from one institution. Consequently, while the results demonstrate technical feasibility, broader deployment across diverse institutional contexts is required to establish generalisability.

The modular architecture of AttendEase ensures high scalability and extensibility, with future expansion vectors including:

- Developing a web administrative panel for managing users, study programmes, schedules, and classrooms;
- Integrating with university information systems to fetch data regarding users, curricula, and schedules;
- Adding features for comparative attendance analysis across semesters, automated alerts for critically low attendance levels, and predictive attendance modelling;
- Optimising processing throughput under high-concurrency user loads.

Acknowledgment

This research was supported by the Scientific Research Fund of Plovdiv University under project MUPD25-FMI-007 'Modern Solutions for Digital Transformation of Business and Education' (2025-2026).

References

[1] T. Fadelelmoula, "The impact of class attendance on student performance," International Research Journal of Medicine and Medical Sciences, vol. 6, no. 2, pp.47-49, 2018.

[2] University of Plovdiv Paisii Hilendarski, Regulation for Educational Activity, 2025, Retrieved from https://uni-plovdiv.bg/uploads/site/pravilnici/Pravilnici2025

[3] A. Kassem, M. Hamad, Z. Chalhoub, and S. El Dahdaah, "An RFID attendance and monitoring system for university applications," In 2010 17th IEEE International Conference on Electronics, Circuits and Systems, pp. 851-854, 2010.

[4] S. Ashok, N. Divya, F. Samuel, and D. Mande, "RFID-based attendance system," Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 12, pp.2239-2248, 2022.

[5] W. Farag, "An RFID-based smart school attendance and monitoring system," BOHR Journal of Computational Intelligence and Communication Network, vol. 1, no. 1, pp. 26-34, 2023.

[6] C. Akinduyite, A. Adetunmbi, O. Olabode, and E. Ibidunmoye, "Fingerprint-based attendance management system," Journal of Computer Sciences and Applications, vol. 1, no. 5, pp. 100-105, 2013.

[7] S. Shah, K. Sharma, R. Gupta, and M. Bhatia, "Face recognition based attendance management system," International Journal for Research in Applied Science and Engineering Technology, vol. 10, pp. 437-439, 2022.

[8] A. Chinedu, O. Grace, and A. Chibuogu, "Quick Response (QR) Code Based Students' Attendance System in the University," International Journal of Research and Innovation in Applied Science, vol. 10, no. 7, pp. 1567-1572, 2025.

[9] E. Karthik, V. Reddy, Y. Gouda, and D. Sivaprashanth, "QR Code Based Attendance Management System," International Journal of Research and Innovation in Social Science, ISSN No. 2454-6186, vol. 9, no. 4, pp. 5173-5220, 2024.

[10] E. Oyerinde, "Design and implementation of a secure QR code-based attendance management system for higher education," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, vol. 11, no. 5, pp. 77-89, 2025.

[11] S. Kurniali, and Mayliana, "The development of a web-based attendance system with RFID for higher education institutions in Binus University," In EPJ Web of Conferences, vol. 68, p. 00038, 2014.

[12] O. Osasuyi, E. Oghogho, and N. Ihedi-Okonkwo, "Design and Implementation of a Fingerprint-Based Attendance Management System," NIPES-Journal of Science and Technology Research, vol. 2, no. 2, 2020.

[13] F. Masalha, and N. Hirzallah, "A student's attendance system using QR code," International Journal of Advanced Computer Science and Applications, vol. 5, no. 3, 2014.

[14] S. Taju, Y. Mamahit, and J. Pongantung, "Implementing QR code and Geolocation Technologies for the Student Attendance System," CogITo Smart Journal, vol. 10, no. 1, pp. 221-232, 2024.

[15] H. Gangurde, R., Dubhalkar, O. Ahire, and P. Borse, "Enhancing Attendance Accuracy with QR Code and Real-Time Location Tracking," International Journal of Advances in Engineering and Management (IJAEM), vol. 7, no. 02, pp. 480- 483, 2025.

[16] M. Muthulakshmi, P. Saravanan, M. Srihari, and P. Shunmugapandian, "Smart QR Code and Geo-Fenced Attendance System," International Journal of Scientific Research & Engineering Trends, vol. 12, no. 2, 2025.

CMST-FL: Cross-Missingness Stable Trust with Evidence-Guided Preprocessing for Federated Healthcare Signal Imputation

A Patient-Disjoint Two-Stage Model for Imputation Selection and Reliability-Weighted Aggregation

Benachir Rigalma, Soumia Ziti, Meryam Belhiah, Hamid Ouhnni
IPSS Team, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

***Abstract*—Missing physiological samples create preprocessing uncertainty and uneven client reliability in federated learning. This study proposes Cross-Missingness Stable Trust federated learning (CMST-FL), a two-stage framework that selects an imputer using development patients and then weights local updates from their marginal validation-quality gains across MCAR, MAR, and contiguous-block masks. Twelve BIDMC records screened five imputers, and 15 disjoint records evaluated five training strategies under three mechanisms, three rates, and three seeds. Iterative imputation minimized the predeclared multi-objective score (0.35) and achieved the lowest standardized imputation RMSE (0.7712) under the tested budget. Across 27 matched evaluation conditions, CMST-FL obtained mean test R^2 of 0.0130, RMSE of 0.3298, and MAE of 0.2624. Compared with FedAvg, its mean R^2 gain was 0.00658, and its RMSE reduction was 0.00110. CMST-FL was favored in all 27 conditions for both metrics, and the paired tests remained statistically significant after Holm correction. Client-level cluster resampling, however, produced wide intervals because only five clients were available, and the worst-client gain was not significant. Trust logs showed meaningful but moderate departures from equal sample weighting. The evidence therefore supports iterative preprocessing followed by CMST-FL as a promising benchmark-specific configuration, while the low absolute R^2 and small error reductions preclude claims of clinical or universal superiority.**

Keywords*—Federated learning; missing data; trust aggregation; physiological signals; multiple imputation; healthcare analytics

I. Introduction

Continuous physiological monitoring produces multivariate signals that are useful for prediction, but practical recordings contain absent or corrupted samples. The BIDMC PPG and Respiration Dataset offers synchronized physiological signals and manually derived respiration targets, making it suitable for controlled studies of missing-data robustness. The present work uses this public resource as a methodological benchmark rather than claiming clinical deployment [1], [2].

Federated learning (FL) enables clients to train a shared model while retaining raw records locally. FedAvg introduced sample-count aggregation for decentralized neural-network training [4]. FedProx later added a proximal local objective for heterogeneous networks [5]. A recent healthcare survey identifies non-identically distributed data, missingness, and uncertain client quality as continuing obstacles [3]. q-FFL addresses fairness across clients [6], whereas FLTrust targets Byzantine updates [7]; neither directly evaluates whether a benign update remains useful across several missingness mechanisms.

This distinction motivates Cross-Missingness Stable Trust federated learning (CMST-FL). Instead of trusting a client because it performs well under only the current mask, CMST-FL asks a counterfactual question: does the local update improve the received global model across a fixed bank of plausible missingness mechanisms? It rewards mean improvement, penalizes cross-mechanism variability and worst-case harm, then smooths trust through rounds before combining it with sample count. The proposed research object is therefore a unified decision model—not only a new server weight—because preprocessing is selected on disjoint development patients before trust aggregation is evaluated.

The study addresses three questions. RQ1 asks which tested imputer provides the best predeclared balance of imputation fidelity, downstream validation performance, and preprocessing time. RQ2 asks whether CMST-FL consistently improves predictive performance over FedAvg and FedProx on patient-disjoint evaluation records. RQ3 asks whether the improvement persists across MCAR, MAR, and contiguous-block missingness at 10%, 30%, and 50% rates.

The main contributions are:

- A new two-stage CMST-FL model that connects leakage-controlled preprocessing selection to trust-weighted aggregation while separating development and evaluation patients;
- A mathematically explicit counterfactual cross-missingness utility with stability, worst-case, and temporal-memory terms requiring only client-side validation summaries and model parameters;
- A matched 27-condition comparison with centralized learning, FedAvg, FedProx and a guarded trust baseline, together with bootstrap intervals and paired Wilcoxon tests; and

- An evidence-bounded interpretation that reports both statistical consistency and the modest absolute magnitude of the observed gains.

The remainder of the study is organized as follows. Section II reviews missing-data and adaptive-aggregation research. Section III defines CMST-FL, Section IV presents the experimental protocol, and Section V reports preprocessing, predictive, cluster-aware, and trust-trajectory results. Section VI interprets the findings, Section VII records the evidence-supported configuration, Section VIII presents limitations, and Section IX concludes the study.

II. Related Work

A. Missing-Data Preprocessing

Simple mean or median replacement is fast but can distort covariance and temporal structure [11]. K-nearest-neighbor imputation estimates absent values from similar observations [8], while iterative imputation models each feature conditionally on the others [9]. Denoising autoencoders learn nonlinear reconstructions and may be attractive for complex physiological data, although their ranking depends strongly on architecture, training budget, and sample size [10]. Missing-data methodology further emphasizes that missingness assumptions and uncertainty should be considered explicitly [11].

Recent federated imputation studies demonstrate growing interest in heterogeneous intensive-care time series and personalized imputation under heterogeneous missingness. CAFE, for example, learns personalized imputation weights from cross-client missing-data heterogeneity. These studies optimize imputation itself, whereas the present work first compares conventional and learned imputers under a fixed budget and then uses the selected imputer inside a reliability-weighted predictive FL pipeline. A method that is best in one dataset or budget cannot be assumed universally best; preprocessing is therefore treated as a patient-disjoint development-stage decision [12], [21].

B. Federated Aggregation and Trust

FedAvg is the sample-weighted reference [4], whereas FedProx changes local optimization through a proximal term [5]. q-FFL prioritizes clients according to loss to improve fairness [6], and FedFomo uses client-specific validation utility [13]. FedLAW learns aggregation weights from client coherence [19]. FedAW introduces dataset-quality-aware weighting [20], while FedLW assigns larger weights to lower-loss clients [22]. These methods show that aggregation need not depend only on sample count, but none evaluates a local update across alternative missingness mechanisms in the manner proposed here.

Robust aggregation and FLTrust principally target anomalous or adversarial updates [7], [14]. Dynamic trust management has also adapted trust thresholds and smoothing parameters according to client performance, behavior, and variability [23]. CMST-FL targets a different reliability question: rather than identifying malicious behavior or using absolute local metrics, it measures the local update's marginal predictive gain relative to the exact received global model, repeats that counterfactual evaluation across a missingness bank, and uses soft sample-aware weights instead of a fixed exclusion decision.

C. Research Gap and Scope

Two gaps are addressed. First, preprocessing and aggregation are often tuned on the same final records, which can overstate generalization. Second, a client weight based on a single validation condition may reward a brittle update. CMST-FL combines patient-disjoint selection with cross-mechanism validation. It does not provide a Byzantine-resilience proof, causal inference, privacy accounting, or clinical certification; these remain outside the present scope.

D. Novelty Positioning

The individual ingredients of CMST-FL—weighted averaging, validation-based quality signals, and temporal smoothing—have precedents. FedLAW studies learned aggregation weights [19], FedAW uses data-quality signals [20], CAFE exploits missing-data heterogeneity for imputation [21], FedLW uses client loss [22], and dynamic trust management adapts trust thresholds and smoothing parameters [23]. To the best of our knowledge, among the methods reviewed in Table I, none combines patient-disjoint imputer selection, local-versus-global gain across a missingness bank, stability and worst-case penalties, and temporally smoothed soft aggregation. The claim is therefore a new integration and formulation, not the invention of trust-weighted FL.

TABLE I. Feature-Level Positioning of CMST-FL Against Close Method Families

Method	Weight/decision signal	Missingness bank	Local–global gain	Stability + worst case	Temporal memory	Imputer selection
FedAvg	Sample count	No	No	No	No	No
FedProx	Sample count + proximal local loss	No	No	No	No	No
FedLAW / FedAW / FedLW	Coherence, data quality or loss	No	No	No	Method-specific	No
CAFE	Personalized imputation weights	Missing-data heterogeneity	No	No	No	Learns imputation
Dynamic trust	Performance and consistency indicators	No	No	No	Yes	No
CMST-FL	Relative gain across mask bank	Yes	Yes	Yes	Yes	Yes

III. Proposed CMST-FL Method

A. System Model

Let K clients hold disjoint patient records. At round t, the server broadcasts global parameters w(t). Client i applies an imputer fitted only on its local training partition, performs one local epoch, and returns updated parameters w_i(t+1). No raw waveform, target value, or missingness mask is transmitted. Each client evaluates the received and locally updated models on the same local validation data under a fixed missingness bank. Fig. 1 summarizes the patient-disjoint two-stage workflow.

B. Cross-Missingness Utility

For each bank condition b in B, Q is the bounded validation-quality score implemented in the experiment. It combines bounded R^2 with scale-normalized RMSE and MAE. For client i, the scale was $s_i = \max(P95(y_i) - P5(y_i), SD(y_i), \varepsilon)$. The following expressions define Q and its normalized terms; the counterfactual gain in Eq. (1) then compares the local update with the exact global model received at that round.

$$Q = \mathrm{clip}(0.50R_\beta^{\,2} + 0.30\exp(-\mathrm{NRMSE}) + 0.20\exp(-\mathrm{NMAE}), 0, 1)$$

$$R_\beta^{\,2} = \{\mathrm{clip}(R^2, -1, 1) + 1\}/2$$

$$\mathrm{NRMSE} = \mathrm{RMSE}/s_i, \quad \mathrm{NMAE} = \mathrm{MAE}/s_i$$

$$g_{i,b}(t) = Q\big(w_i(t+1); V_{i,b}\big) - Q\big(w(t); V_{i,b}\big). \tag{1}$$

The experiment used B = {MCAR 30%, MAR 30%, block 30%}. A useful update should have positive average gain, low variation across mechanisms, and no severely negative condition. The penalized utility is

$$u_i(t) = \underset{b\in\mathcal{B}}{\mathrm{mean}}\big[g_{i,b}(t)\big] - \lambda \underset{b\in\mathcal{B}}{\mathrm{sd}}\big[g_{i,b}(t)\big] - \mu\max\Big(0, -\min_{b\in\mathcal{B}} g_{i,b}(t)\Big). \tag{2}$$

Here λ controls the stability penalty and μ penalizes worst-case harm. Unlike a weight based on one absolute local metric, the gain is relative to the exact global model received at that round. This construction reduces persistent preference for clients with intrinsically easier validation records.

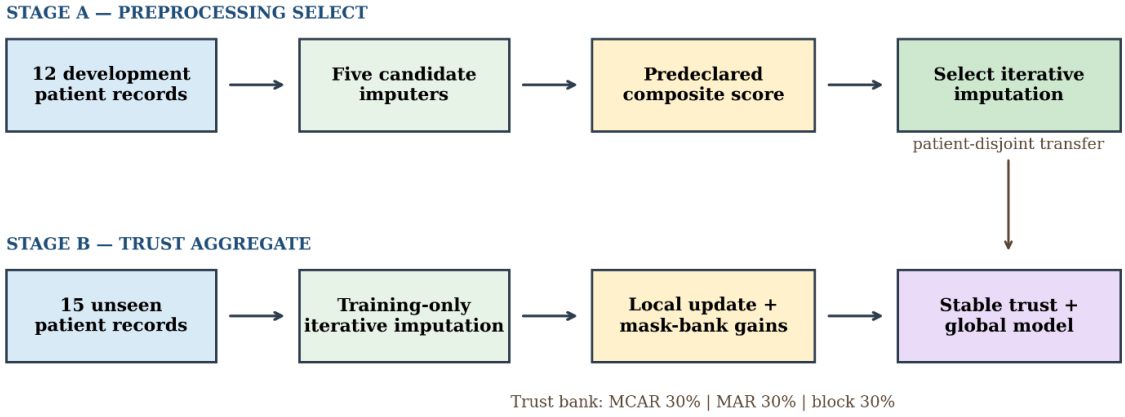


Fig. 1. Unified two-stage CMST-FL model. Stage A selects the imputer on development patients; Stage B transfers only the selected preprocessing decision to unseen patients and performs cross-missingness stable-trust aggregation.

C. Temporal Trust and Aggregation

The instantaneous trust is a temperature-scaled logistic transformation. Exponential smoothing reduces round-to-round weight volatility:

$$z_i(t) = \frac{1}{1+\exp\left(-\frac{u_i(t)}{\tau}\right)}, \tag{3}$$

$$T_i(t) = \rho T_i(t-1) + (1-\rho) z_i(t). \tag{4}$$

CMST-FL combines trust with the number n_i of local training examples. With trust floor epsilon and concentration exponent gamma, the normalized aggregation coefficient is

$$a_i(t) = \frac{n_i[\varepsilon + T_i(t)]^\gamma}{\sum_{j=1}^{K} n_j[\varepsilon + T_j(t)]^\gamma}, \tag{5}$$

$$w(t+1) = \sum_{i=1}^{K} a_i(t) w_i(t+1). \tag{6}$$

The floor prevents a client from receiving zero weight, whereas γ controls how strongly reliable clients are emphasized. FedAvg is recovered conceptually when the trust multiplier is constant across clients. Parameters were fixed before Stage B: λ=0.5, μ=1.0, τ=0.05, ρ=0.8, ε=0.05, γ=2, and $T_i(0)$=0.5.

D. Integrated Two-Stage Decision Model

The complete contribution is represented as an ordered model rather than an isolated aggregation equation. Let I* be the development-selected imputer, N_train the training-only normalization operator, B the validation-mask bank, T(T) the final client-trust vector, and w(T) the final global parameters. The model returned by the procedure is

$$\begin{aligned} \mathcal{M}_{\mathrm{CMST}} &= \{I^*, N_{\mathrm{train}}, \mathcal{B}, T(T), w(T)\}, \\ I^* &= \underset{j}{\mathrm{argmin}}\ S_j. \end{aligned} \tag{7}$$

This representation makes the deployment decision explicit: the imputer and normalization are fixed before evaluation patients are exposed, whereas trust and model parameters evolve only inside Stage B. Table II converts the mathematical definition into five operational steps and their outputs.

TABLE II. OPERATIONAL DEFINITION OF THE TWO-STAGE CMST-FL MODEL

Stage	Input	Decision rule	Output
A—Screen	Development patients; five imputers	Minimize predeclared score S_j	Selected imputer I*
B—Prepare	Unseen patient partitions	Fit I* and normalization on training only	Leakage-controlled client data
B—Evaluate	Global and local models; bank B	Compute relative gains and penalized utility	Instantaneous trust
B—Stabilize	Current and previous trust	Logistic mapping and temporal smoothing	T_i(t)
B—Aggregate	Local models, samples and trust	Sample-aware soft weighting	Final global model w(T)

IV. EXPERIMENTAL DESIGN

A. Dataset, Features and Patient Split

The experiment used BIDMC CSV records derived from the PhysioNet dataset [1], [2]. RESP was the regression target, and PLETH, lead II, lead V, and aVR were predictors. Twenty-seven patient recordings were selected deterministically: 12 development records for preprocessing screening and 15 non-overlapping evaluation records for the aggregation study. Each record was uniformly subsampled to at most 200 rows to respect the two-day compute budget. Within every record, the first 60% of samples formed the training partition, the next 20% validation, and the final 20% test, preserving temporal order.

Five clients received patient-record partitions created with a fixed seed. Training-only pooled sufficient statistics were used for feature and target standardization. This design prevents a test statistic, future temporal segment, or evaluation patient from influencing imputer fitting, normalization, preprocessing selection, or trust calculation.

B. Missingness and Preprocessing Screen

Synthetic masks were generated per patient record and seed. MCAR independently masked entries. MAR selected missing rows using a logistic propensity derived from another feature. Block missingness created contiguous missing segments. A safeguard retained at least one observed feature per row. Native missing values were preserved and were excluded from the synthetic-mask imputation metric.

Stage A compared mean, median, distance-weighted five-neighbor KNN, iterative chained regression (10 iterations, median initialization), and a denoising autoencoder. The autoencoder used two ReLU hidden layers, Adam, and three epochs; therefore, its result represents only the tested low-budget configuration. Screening comprised three mechanisms at 30% missingness and seeds 11 and 22. Each method was evaluated with centralized learning and five-round FedAvg.

The predeclared score S_j combined percentile ranks, with lower values better. R^2 ranks were reversed so that higher R^2 received a lower rank:

$$S_j = 0.35r_{\mathrm{imp},j} + 0.20r_{\mathrm{F,RMSE},j} + 0.10r_{\mathrm{F},R^2,j} + 0.15r_{\mathrm{C,RMSE},j} + 0.05r_{\mathrm{C},R^2,j} + 0.15r_{\mathrm{time},j}. \quad (8)$$

Thus, direct imputation carried 35% weight, downstream federated validation 30%, centralized validation 20%, and log-transformed preprocessing time 15%. Only the winning method was advanced to Stage B. Table III consolidates the patient split, model budget, missingness conditions, and fixed aggregation settings.

TABLE III. REPRODUCIBLE EXPERIMENTAL PROTOCOL

Item	Specification
Development records	12 patient records; no Stage B patients
Evaluation records	15 disjoint patient records
Clients	5
Rows per record	At most 200, uniformly spaced
Split	60% train / 20% validation / 20% test, chronological
Stage A screen	5 imputers × 3 mechanisms × 1 rate × 2 seeds
Stage B evaluation	5 strategies × 3 mechanisms × 3 rates × 3 seeds
Central / FL training	10 epochs / 10 rounds; 1 local epoch per round
Network	4–128–64–1 multilayer perceptron; ReLU; Adam 0.001
FedProx	Proximal coefficient 0.01
CMST bank	MCAR, MAR, and block, each at 30%
Seeds	Stage A: 11, 22; Stage B: 11, 22, 33
Execution platform	64-bit Windows 10 Pro laptop; Intel Core i3-5005U CPU at 2.00 GHz; 2 physical cores and 4 logical processors; 7.9 GB RAM; Intel HD Graphics 5500; TensorFlow detected no GPU accelerator, so execution was CPU-only.
Software constraints	Flower ≥1.12,<1.13; TensorFlow ≥2.16,<2.19; NumPy ≥1.26,<2.0; scikit-learn ≥1.4,<2.0; SciPy ≥1.11,<2.0

C. Aggregation Baselines

Stage B compared: 1) centralized learning with the same architecture and total epoch count; 2) FedAvg; 3) FedProx with proximal coefficient 0.01; 4) CMST-FL; and 5) FedGuard, an experimental server-side selector that interpolated sample weights and trust/alignment weights using candidate coefficients 0, 0.25, and 0.50 subject to validation gates. FedGuard is treated as a secondary guarded baseline, while CMST-FL is the proposed method.

All FL methods ran for 10 synchronous rounds with one local epoch, all five clients participating and failures disallowed. Strategy comparisons reused the same patient partitions, masks, and seeds. The implementation used Flower for federated communication [18] and TensorFlow/Keras for the multilayer perceptron [24].

D. Metrics and Statistical Analysis

Primary predictive metrics were test R^2, RMSE, and MAE; higher R^2 and lower errors are better [15], [16]. Imputation RMSE and MAE were computed only at synthetically masked standardized feature entries. We defined worst-client R^2 as the minimum local test R^2 among the five clients for each federated method. Centralized worst-client entries were not used for fairness conclusions because the centralized evaluator produced a pooled metric rather than five local metrics.

Means were calculated across conditions, and 95% percentile bootstrap confidence intervals were estimated with 5,000 resamples [25]. Paired analyses used the 27 matched mechanism–rate–seed conditions and two-sided Wilcoxon signed-rank tests [26]. Holm correction controlled familywise error across the reported strategy–metric comparisons [17]. Because those 27 conditions reuse the same five clients, a supplementary client-cluster bootstrap resampled clients with replacement (50,000 resamples; seed 20260913). The cluster analysis is reported separately and is interpreted cautiously because five clusters cannot support precise population-level uncertainty.

V. Results

A. Preprocessing Selection

Table IV and Fig. 2 summarize Stage A. Iterative imputation achieved the lowest selection score (0.35) and the best direct imputation RMSE (0.7712). KNN had slightly better FedAvg validation RMSE and R^2, but weaker direct imputation error and centralized validation performance. Mean and median replacement were fastest. The DAE required 3.42 s and had weaker FedAvg validation performance under its three-epoch budget. Because no larger DAE budget was run, this outcome excludes only the tested configuration and does not establish general DAE inferiority.

TABLE IV. Stage-A Preprocessing Screen (Six Development Conditions)

Method	Imp. RMSE	Central R^2	FedAvg R^2	Time (s)	Score
Iterative	0.7712	0.1641	0.0846	0.273	0.35
KNN	0.9400	0.1341	0.0920	0.226	0.56
Mean	0.9515	0.1614	0.0744	0.078	0.57
DAE	0.8605	0.1539	0.0320	3.416	0.75
Median	0.9808	0.1603	0.0637	0.079	0.77

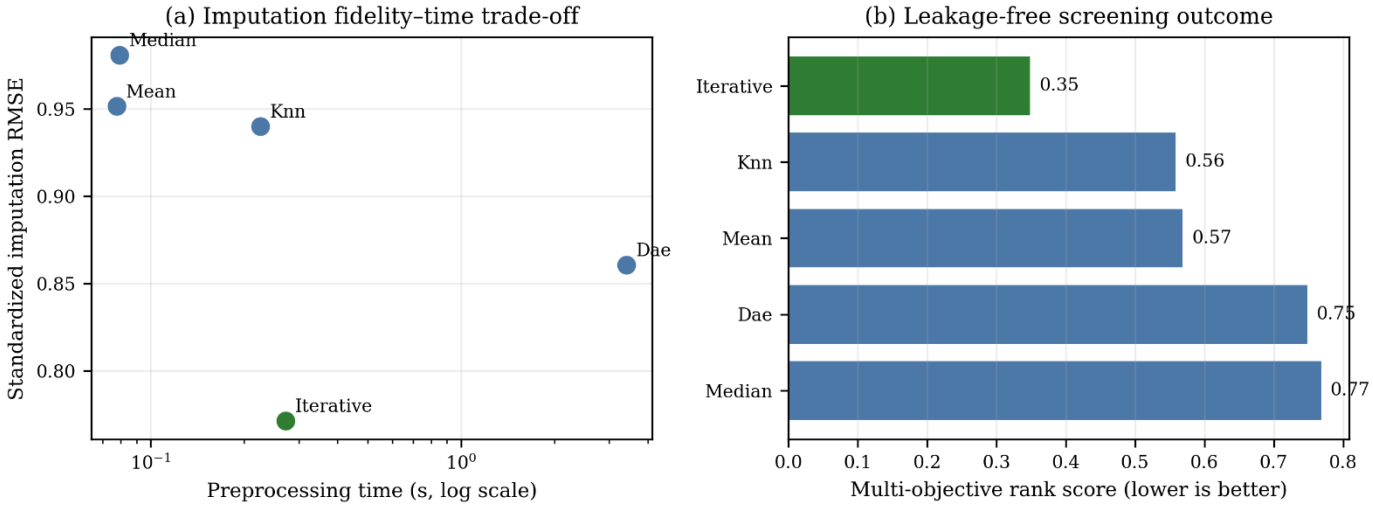


Fig. 2. Preprocessing selection. Iterative imputation provided the best predeclared multi-objective score under the tested data and compute budget; the result is not a universal ranking of imputers.

B. Overall Aggregation Performance

Table V summarizes the 27 Stage B runs per strategy, and Fig. 3 shows the corresponding means and bootstrap intervals. CMST-FL obtained the highest mean R^2 (0.0130) and the lowest RMSE (0.3298) and MAE (0.2624). Relative to FedAvg, its average R^2 gain was 0.00658, while RMSE and MAE decreased by 0.00110 (0.33%) and 0.00091 (0.35%), respectively. The directions were consistent, but the absolute differences were small.

TABLE V. STAGE-B TEST PERFORMANCE; MEAN AND 95% BOOTSTRAP CI FOR R^2

Strategy	R^2 [95% CI]	RMSE	MAE	Worst-client R^2
Centralized	0.0045 [-0.0007, 0.0098]	0.3312	0.2649	—
FedAvg	0.0064 [0.0037, 0.0094]	0.3309	0.2633	-0.0738
FedProx	0.0061 [0.0033, 0.0091]	0.3310	0.2627	-0.0725
FedGuard	0.0077 [0.0052, 0.0104]	0.3307	0.2631	-0.0673
CMST-FL	0.0130 [0.0104, 0.0157]	0.3298	0.2624	-0.0735

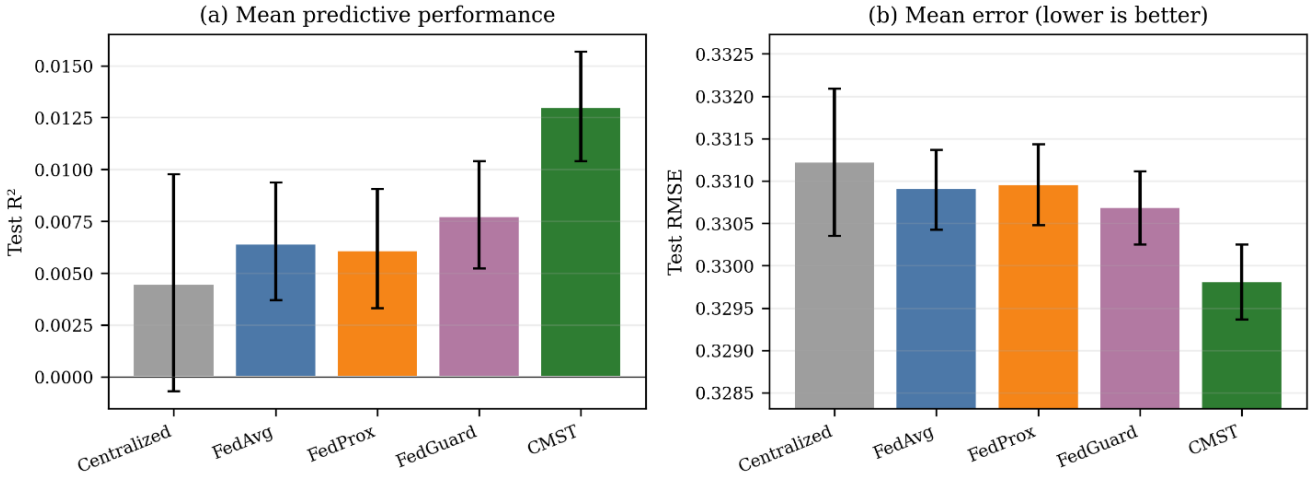


Fig. 3. Mean Stage B performance across 27 matched conditions. Error bars are 95% bootstrap confidence intervals. The truncated RMSE axis is used to make small between-method differences visible.

C. Paired Comparison with FedAvg

Table VI reports paired results. CMST-FL improved R^2 and RMSE in all 27 matched conditions and improved MAE in 25. Holm-adjusted p-values were below 0.000001 for those three metrics, and condition-level intervals excluded zero. In contrast, the mean worst-client R^2 gain was 0.00027, its interval crossed zero, and the adjusted p-value was 1.0. Thus, the experiment supports a small pooled predictive improvement, not a reliable fairness improvement.

D. Dependence-Aware and Trust-Trajectory Analysis

The client-cluster bootstrap retained the 135 matched client–condition differences but resampled the five client clusters. The mean client-level R^2 improvement was 0.00594, with a 95% cluster interval of [-0.01229, 0.03037]. The mean RMSE reduction was 0.00102, with an interval of [-0.00207, 0.00518]. Both intervals crossed zero. This result does not contradict the condition-level paired test; it shows that uncertainty over a broader client population remains large when only five clients are observed.

Table VII and Fig. 4 connect CMST-FL's utility calculation to the aggregation trajectory. Across 1,350 client–round rows, temporal trust ranged from 0.361 to 0.745, and normalized weights ranged from 0.087 to 0.306. The mean absolute departure from the equal 0.20 weight was 0.034; penalized utility and weight were strongly associated (r=0.918).

TABLE VI. PAIRED CMST-FL IMPROVEMENT OVER FEDAVG (27 MATCHED CONDITIONS)

Metric	Mean improvement	95% bootstrap CI	Win rate	Holm p
R^2 gain	0.006581	[0.005287, 0.007907]	100.0%	<0.000001
RMSE reduction	0.001098	[0.000882, 0.001318]	100.0%	<0.000001
MAE reduction	0.000910	[0.000672, 0.001146]	92.6%	<0.000001
Worst-client R^2 gain	0.000267	[-0.006561, 0.007737]	40.7%	1.000

TABLE VII. CLIENT UTILITY, TEMPORAL TRUST AND NORMALIZED CMST-FL WEIGHT

Client	Raw utility	Penalized utility	Mean trust	Mean weight	Weight range
C1	0.0274	0.0256	0.5806	0.2147	[0.1796, 0.2449]
C2	0.0227	0.0210	0.5702	0.2078	[0.1872, 0.2310]
C3	-0.0062	-0.0178	0.4461	0.1343	[0.0867, 0.1960]
C4	0.0499	0.0485	0.6452	0.2609	[0.2039, 0.3055]
C5	0.0107	0.0087	0.5307	0.1823	[0.1631, 0.2066]

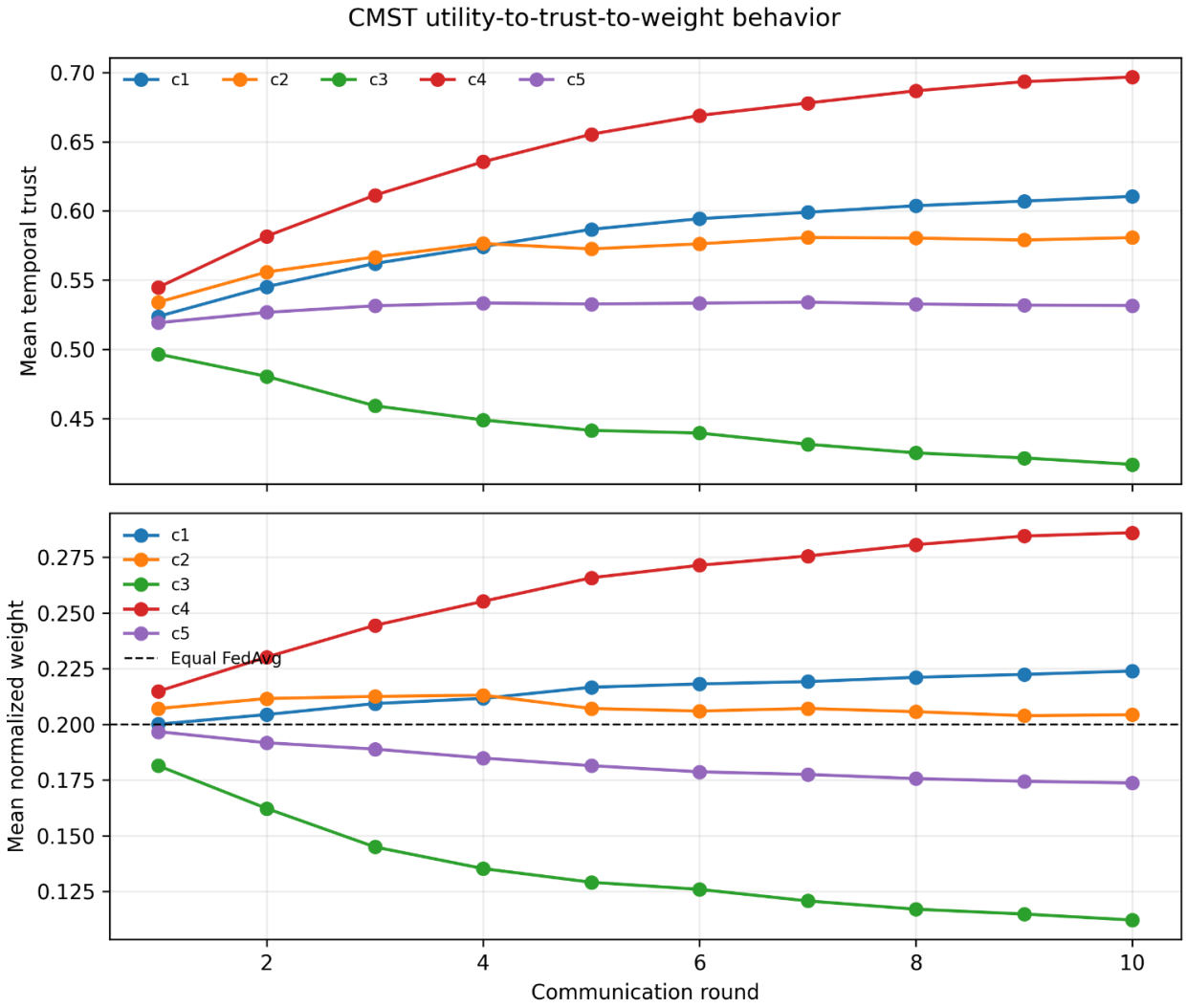


Fig. 4. CMST-FL temporal trust and normalized aggregation-weight trajectories, averaged over the 27 evaluation conditions. The dashed weight reference is the equal 0.20 allocation used because every client had 360 training examples.

Client C4 received the largest mean weight (0.261), whereas C3 received the smallest (0.134), confirming that CMST-FL materially changed the optimization weights. Nevertheless, all clients had 360 training examples, so this experiment isolates the trust multiplier and does not empirically validate behavior under unequal sample counts.

E. Robustness Across Missingness Conditions

Fig. 5 and Table VIII report performance across the held-out evaluation grid. The CMST-FL R^2 gain over FedAvg was positive in every mechanism–rate cell after averaging three seeds. The largest mean gain occurred for MCAR at 30% (+0.01107), while the smallest occurred for block missingness at 50% (+0.00281). CMST-FL also had the highest R^2 among the four federated strategies in each of the 27 conditions. Because the validation bank used the same mechanism families, these results demonstrate robustness within MCAR, MAR, and block missingness—not generalization to genuinely unseen mechanisms.

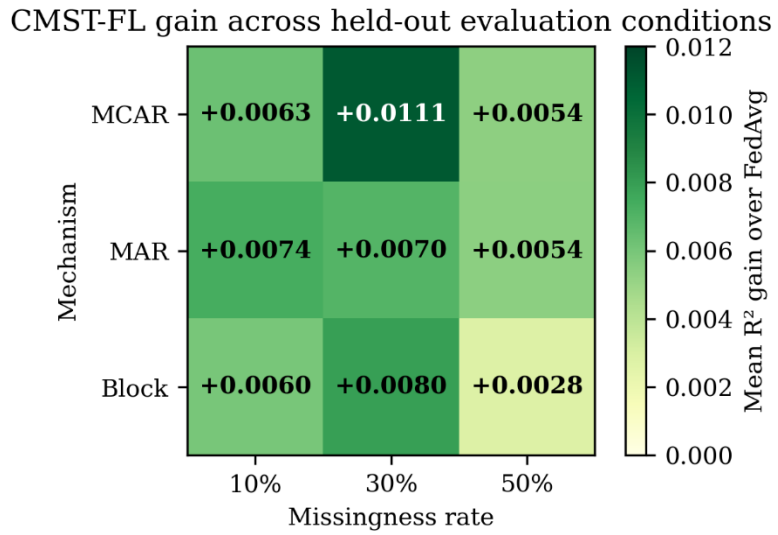


Fig. 5. Mean CMST-FL test R^2 gain over FedAvg by missingness mechanism and rate. Every cell is positive; each cell averages three seeds.

TABLE VIII. MEAN TEST R^2 BY MISSINGNESS CONDITION (THREE SEEDS)

Mechanism	Rate	Centralized	FedAvg	CMST-FL
MCAR	10%	0.0052	0.0097	0.0160
MCAR	30%	0.0065	0.0039	0.0149
MCAR	50%	0.0019	0.0047	0.0100
MAR	10%	0.0131	0.0065	0.0139
MAR	30%	0.0108	0.0074	0.0144
MAR	50%	0.0008	0.0064	0.0118
Block	10%	0.0166	0.0133	0.0193
Block	30%	0.0017	0.0049	0.0129
Block	50%	-0.0164	0.0009	0.0037

F. Centralized–Federated Gap

CMST-FL exceeded the matched centralized R^2 and reduced matched RMSE in 18 of 27 conditions. Averaged over all conditions, CMST-FL improved R^2 by 0.00851 and reduced RMSE by 0.00141 relative to the centralized reference. This result should not be read as a general advantage of FL: the centralized model has access to pooled data but was constrained to the same compact architecture, ten epochs, and a reduced dataset. The defensible conclusion is that CMST-FL reduced and, in this benchmark, slightly reversed the average centralized–federated gap under the fixed budget.

VI. DISCUSSION

A. Why CMST-FL Helped

Three design elements plausibly explain the consistency. First, trust is based on improvement relative to the received model, rather than absolute local performance. Second, the bank discourages an update that is beneficial under one mask but harmful under another. Third, temporal smoothing limits abrupt reweighting from a single noisy validation round. The sample-count factor retains FedAvg's preference for estimates supported by more local examples.

The results also show why significance alone is insufficient. All 27 paired R^2 and RMSE differences favored CMST-FL, producing very small p-values, yet the absolute error reduction was about one-thousandth. This is encouraging evidence of a stable directional effect, not evidence of a large clinical benefit. The low absolute R^2 values indicate that the compact four-input model explains little test variance and that the task remains difficult after aggressive subsampling.

B. Preprocessing Interpretation

Iterative imputation was preferred because it combined the lowest direct imputation error with strong centralized validation results and moderate runtime. It did not lead to the best individual downstream metric: KNN achieved slightly stronger FedAvg validation R^2. The multi-objective score makes this trade-off explicit. Therefore, the article can answer which tested method was best under the defined score and budget, but it should not claim that iterative imputation is universally superior. Longer DAE training, different signal windows, or richer temporal models could change the ranking.

C. Comparison with FedProx and Guarded Weighting

FedProx produced nearly the same mean R^2 as FedAvg under the single fixed coefficient μ=0.01 and one local epoch.

Because no coefficient sweep was performed, the result characterizes only this configuration and cannot establish a general limitation of proximal regularization. FedGuard improved mean R^2 over FedAvg but remained below CMST-FL. Its validation gate accepted a nonzero trust mixture in 38 of 270 server rounds (14.1%), so most rounds retained sample weighting.

D. Practical Implications

CMST-FL is most applicable when each participating institution can reserve a local validation segment and evaluate client updates under several plausible missingness conditions without sharing raw patient data. Compared with FedAvg, the additional client-side workload consists mainly of validation inference under the three conditions in the missingness bank. The server then calculates the utility, updates the temporal trust values, and normalizes the aggregation weights. These server-side operations are lightweight relative to local model training. However, institutions must maintain representative local validation data, and the choice of missingness conditions should reflect the types of signal loss expected in the intended application. The archived experiment controller recorded a total runtime of 5.081 hours for the complete two-stage pipeline. The experiment was executed on a 64-bit Windows 10 Pro laptop equipped with an Intel Core i3-5005U CPU operating at 2.00 GHz, two physical cores, four logical processors, and 7.9 GB of RAM. Although the laptop included Intel HD Graphics 5500, TensorFlow detected no GPU accelerator; consequently, the experiment used CPU-only execution. The hardware specifications were verified on the same machine after the experiment was completed. The reported runtime should therefore be interpreted as an observed result for this specific hardware and experimental configuration rather than as a hardware-normalized performance benchmark. Nevertheless, completing the pipeline on modest hardware suggests that the proposed framework can be evaluated without specialized high-performance computing resources. Its practical value remains subject to confirmation using larger client populations, longer training schedules, and full-resolution physiological signals.

VII. Evidence-Supported Final Model

The experiment supports one concrete model for the present benchmark: training-only iterative imputation followed by CMST-FL aggregation. The decision was not selected from the final test results. Iterative imputation was frozen after Stage A because it minimized the development score; the complete CMST configuration was then evaluated on disjoint patients. Table IX records the resulting model so that the contribution is reproducible and falsifiable.

TABLE IX. Final Empirical Configuration of the CMST-FL Model

Component	Selected specification	Evidence or role
Preprocessing	Iterative chained regression; 10 iterations; median initialization	Best composite score 0.35 and lowest imputation RMSE 0.7712
Normalization	Training-only pooled sufficient statistics	Prevents validation/test leakage
Predictor	PLETH, II, V and aVR → RESP; dense 128–64–1 network	Same compact architecture across strategies
Trust evidence	Relative local-versus-global composite quality gain under MCAR, MAR and block at 30%	Measures cross-mechanism marginal utility
Stability	Mean gain − 0.5 SD − worst-case negative-gain penalty	Discourages brittle or harmful updates
Temporal memory	ρ=0.8; logistic temperature 0.05; initial trust 0.5	Reduces round-to-round volatility
Aggregation	$n_i(0.05 + T_i)^2$ normalized across clients	Equal n_i=360 here; unequal-sample behavior remains untested
Observed outcome	R^2 0.0130; RMSE 0.3298; MAE 0.2624	Best mean pooled metrics among tested strategies

The strongest result is consistency rather than magnitude: CMST-FL improved R^2 and RMSE over FedAvg in every matched condition, yet the absolute error reduction was small. Consequently, the final model should be presented as an evidence-supported prototype that narrows the tested federated performance gap, not as a guaranteed optimal or clinically validated model.

VIII. Threats to Validity and Limitations

The primary limitation is scale. The evaluation used 27 of the 53 BIDMC records, each restricted to 200 uniformly spaced rows, five clients, and one dataset. Training stopped after 10 synchronous rounds, so the ranking is a fixed-budget comparison rather than evidence of convergence. Larger client populations, longer training, full-resolution windows, and external datasets are required before population-level or clinical claims can be made.

Second, missingness was synthetically injected. MCAR, propensity-based MAR, and contiguous blocks represent different patterns, but real sensor loss may be nonstationary, device-specific, or missing-not-at-random. The validation bank used the same three families as the evaluation grid, so genuinely unseen mechanisms were not tested. Third, the small feed-forward predictor did not model temporal windows, which helps explain the low R^2 values. Fourth, all clients had equal sample counts, leaving the sample-aware factor untested under quantity heterogeneity.

Fifth, Stage A used two seeds and Stage B three seeds. The client-cluster analysis exposes the resulting uncertainty but cannot overcome having only five clusters. The DAE received only three epochs, FedProx used only μ=0.01, and CMST-FL hyperparameters were fixed. No component ablation removed the stability, worst-case, temporal, or nonlinear-weighting terms; consequently, the current experiment supports the complete formulation but cannot attribute the gain to an individual component. Hardware specifications were verified after execution on the same machine, although separate wall-clock times and communication overhead for the individual

strategies were not recorded. Future work will address these limitations with DAE and FedProx sensitivity analyses, incremental CMST-FL ablations, at least 30 rounds, more clients, unseen mask families, unequal sample sizes, and stronger temporal predictors.

IX. Conclusion

This study introduced CMST-FL as a two-stage preprocessing-and-aggregation framework for incomplete physiological signals. Its distinguishing formulation evaluates each client update against the received global model with a composite quality score across several missingness mechanisms, penalizes unstable and worst-case harmful gains, and smooths trust across rounds. Stage A selected iterative imputation under the predeclared multi-objective score; Stage B found that CMST-FL improved R^2 and RMSE over FedAvg in all 27 matched conditions and achieved the best mean pooled metrics among the tested strategies.

The evidence therefore supports iterative training-only preprocessing followed by cross-missingness stable-trust aggregation for this benchmark. The novelty claim is specific to this integration and formulation. The low absolute R^2, small error reductions, wide client-cluster intervals, fixed 10-round budget, and lack of ablations preclude claims of universal or clinical superiority.

Reproducibility and Data Availability

The BIDMC PPG and Respiration Dataset is publicly available from PhysioNet subject to its terms [1], [2]. The experimental archive preserves configuration files, deterministic seeds, patient partitions, missingness masks generated from stable identifiers, raw per-run JSON/CSV logs, console logs, and scripts used to create the publication tables. An anonymized code-and-results archive can be released with the manuscript. No new patient data were collected.

Evidence and figure provenance: Reported experimental values and Fig. 2, 3, and 5 were computed from the archived Stage A and Stage B CSV/JSON outputs. Fig. 4 was computed from the archived per-client utility, trust, and weight logs. Fig. 1 is a conceptual workflow. All figures were created programmatically; no generative-image model was used. Displayed values are rounded, and intervals and p-values are derived statistics rather than raw observations.

Declarations

Conflict of interest: The authors declare no conflict of interest. Ethical statement: The study used an openly available, de-identified dataset and involved no new human-participant data collection. Data availability: The source dataset is publicly available through PhysioNet [1], [2]; the analysis code and derived result tables can be supplied as supplementary material.

AI-assisted writing disclosure: A generative-AI assistant supported language editing, organization, equation typesetting, formatting, and evidence cross-checking; it did not generate experimental observations or numerical results. The authors remain responsible for independently verifying the code, logs, calculations, citations, claims, and final text.

References

[1] M. A. F. Pimentel, A. E. W. Johnson, P. H. Charlton, D. Birrenkott, P. J. Watkinson, L. Tarassenko, and D. A. Clifton, "Toward a robust estimation of respiratory rate from pulse oximeters," IEEE Trans. Biomed. Eng., vol. 64, no. 8, pp. 1914–1923, 2017, doi: 10.1109/TBME.2016.2613124.

[2] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," Circulation, vol. 101, no. 23, pp. e215–e220, 2000, doi: 10.1161/01.CIR.101.23.e215.

[3] M. Ali, F. Naeem, M. Tariq, and G. Kaddoum, "Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey," IEEE J. Biomed. Health Inform., vol. 27, no. 2, pp. 778–789, 2023, doi: 10.1109/JBHI.2022.3181823.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. AISTATS, PMLR, vol. 54, pp. 1273–1282, 2017.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in Proc. MLSys, 2020.

[6] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in Proc. ICLR, 2020.

[7] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," in Proc. NDSS, 2021, doi: 10.14722/ndss.2021.24434.

[8] O. Troyanskaya et al., "Missing value estimation methods for DNA microarrays," Bioinformatics, vol. 17, no. 6, pp. 520–525, 2001, doi: 10.1093/bioinformatics/17.6.520.

[9] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," J. Stat. Softw., vol. 45, no. 3, pp. 1–67, 2011, doi: 10.18637/jss.v045.i03.

[10] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proc. ICML, pp. 1096–1103, 2008, doi: 10.1145/1390156.1390294.

[11] R. J. A. Little and D. B. Rubin, Statistical Analysis with Missing Data, 3rd ed. Hoboken, NJ, USA: Wiley, 2019.

[12] R. Vavekanand, A. A. Sathio, and M. Sultani, "A federated learning framework for deep imputation of missing data in heterogeneous ICU time series," Sci. Rep., vol. 16, Art. no. 18097, 2026, doi: 10.1038/s41598-026-49311-7.

[13] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first-order model optimization," in Proc. ICLR, 2021.

[14] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," IEEE Trans. Signal Process., vol. 70, pp. 1142–1154, 2022, doi: 10.1109/TSP.2022.3153135.

[15] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput. Sci., vol. 7, Art. no. e623, 2021, doi: 10.7717/peerj-cs.623.

[16] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not," Geosci. Model Dev., vol. 15, pp. 5481–5487, 2022, doi: 10.5194/gmd-15-5481-2022.

[17] S. Holm, "A simple sequentially rejective multiple test procedure," Scand. J. Stat., vol. 6, no. 2, pp. 65–70, 1979.

[18] D. J. Beutel et al., "Flower: A friendly federated learning research framework," arXiv:2007.14390, 2020.

[19] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in Proc. 40th Int. Conf. Mach. Learn. (ICML), PMLR, vol. 202, pp. 19767–19788, 2023.

[20] Y. Tang, "Adapted weighted aggregation in federated learning," Proc. AAAI Conf. Artif. Intell., vol. 38, no. 21, pp. 23763–23765, 2024, doi: 10.1609/aaai.v38i21.30557.

[21] S. Min, H. Asif, X. Wang, and J. Vaidya, "CAFE: Improved federated data imputation by leveraging missing data heterogeneity," IEEE Trans.

Knowl. Data Eng., vol. 37, no. 5, pp. 2266–2281, 2025, doi: 10.1109/TKDE.2025.3537403.

[22] C. Yao, H. Wang, F. Yang, Y. Yang, and Z. Guo, "A loss-based weighted aggregation method for federated learning with heterogeneous computing resources," in Proc. 9th Asia-Pacific Workshop on Netw. (APNet), pp. 307–309, 2025, doi: 10.1145/3735358.3737773.

[23] P. Shepherd, T. Dagiuklas, B. Alkan, and J. Rodriguez, "Dynamic trust management for secure federated learning in critical industrial and IoT networks," in Proc. IEEE ICC Workshops, pp. 371–376, 2025, doi: 10.1109/ICCWorkshops67674.2025.11162455.

[24] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI), pp. 265–283, 2016.

[25] B. Efron, "Bootstrap methods: Another look at the jackknife," Ann. Stat., vol. 7, no. 1, pp. 1–26, 1979, doi: 10.1214/aos/1176344552.

[26] F. Wilcoxon, "Individual comparisons by ranking methods," Biometrics Bull., vol. 1, no. 6, pp. 80–83, 1945, doi: 10.2307/3001968.

A Fairness–Utility Evaluation Framework for Assessing Large Language Models (LLMs)

Samah Alhazmi
Computer Science Department, College of Computing and Informatics,
Saudi Electronic University, Riyadh 11673, Saudi Arabia

***Abstract*—Large language models are increasingly used in contexts where their outputs can affect people directly, including hiring, admissions, and lending. This growing role makes it important to consider not only how well these models perform, but also whether their behavior is fair. Although many fairness metrics, bias benchmarks, and mitigation techniques have been proposed, there is still limited guidance on how fairness and predictive performance can be evaluated together through a consistent procedure. This study presents a five-stage Fairness–Utility Evaluation Framework that brings together dataset selection and preparation, model assessment, fairness evaluation, utility evaluation, and trade-off analysis and reporting. Rather than introducing a new metric, the framework organizes these elements into a structured and reproducible evaluation process. Its application is illustrated using LLaMA-2–7B, CrowS-Pairs, and WinoBias to show how the five stages can be followed in practice; the example is intended to demonstrate the procedure rather than provide new experimental results. Documentation is incorporated throughout the process so that evaluation choices, measures, and outcomes can be traced and reviewed. The resulting framework provides researchers and professionals with a structured way to consider fairness and performance together when evaluating an LLM for potential deployment.**

***Keywords*—Large language models; fairness–utility evaluation; AI governance; natural language processing**

I. Introduction

Large language models are no longer limited to research environments. They are increasingly used in education, employment, and other contexts where their outputs may affect people directly [1]. Their broad language capabilities have supported this wider use.

Fairness has consequently become an important concern in the evaluation of LLMs. LLMs are trained on large collections of text that may contain historical stereotypes and demographic biases [2], [3]. Models can learn these patterns during training and reproduce them in their outputs, which may lead to biased outcomes when the models are applied.

Several approaches have been developed to reduce bias in LLMs. These methods can be applied to the training data, during model training, or to the outputs after generation. However, improving fairness may also affect predictive performance [4]. For this reason, evaluating a model requires considering fairness together with its performance on the intended task.

Recent studies have examined fairness from several perspectives. A survey of 121 studies [5] examined bias sources as well as methods for detecting, evaluating, and mitigating bias. Fairness–utility relationships have also been examined in text-to-image generation [6], while another study investigated how LLMs respond to fairness-related questions in resource-allocation tasks [7]. Despite this progress, studies still vary in how fairness and utility are evaluated and reported. A consistent and reproducible procedure for assessing both within the same evaluation remains limited.

This study addresses that gap. Much of the existing LLM fairness literature focuses on individual metrics, mitigation techniques, or specific models. Less attention has been given to combining these elements within the same evaluation process. Documentation also receives limited attention, even though it is needed to understand how an evaluation was conducted and how its findings were reached. Bringing fairness, utility, and documentation into one framework could provide a clearer basis for evaluating an LLM before deployment.

To address this need, this study proposes a Fairness–Utility Evaluation Framework. The framework follows a common sequence that begins with benchmark selection and preparation, then moves through model assessment, fairness and utility evaluation, and trade-off analysis and reporting. The procedure can be applied across different LLMs and application contexts.

Documentation is maintained throughout the framework and is not limited to the final reporting stage. Recording the evaluation procedure, key choices, and supporting evidence makes it possible to review how the assessment was conducted. Considering fairness and predictive performance within the same process also helps evaluators use both types of evidence when making decisions about LLM deployment.

A. Research Contributions

The main contributions of this study are as follows:

- A Fairness–Utility Evaluation Framework organized into five stages, covering the evaluation process from dataset preparation to trade-off analysis and reporting.
- An approach that evaluates fairness and utility together, allowing their results to be interpreted within the same assessment.
- A model- and benchmark-independent structure that can be used with different LLMs, benchmarks, and application contexts.
- An illustrative example using established fairness benchmarks to demonstrate how the framework can be applied in practice.

These contributions provide a common structure for evaluating fairness and utility while incorporating documentation relevant to AI governance.

B. Research Questions

Three research questions guide the study:

- RQ1: What components should a comprehensive fairness–utility evaluation framework include?
- RQ2: How can fairness assessment and predictive-performance evaluation be integrated into one unified, reproducible workflow?
- RQ3: How can documentation and governance-oriented reporting be built into that workflow to support transparency and accountability in LLM deployment?

The remainder of the study is organized as follows. Section II reviews prior work on bias, fairness evaluation, and governance in LLMs. Section III presents the framework and its five stages, followed by the illustrative example in Section IV. Section V discusses the implications and limitations of the proposed framework, and Section VI concludes the study.

II. Related Work

A. Bias in Large Language Models

Concerns about bias in language models are not new. Early research identified social stereotypes in word embeddings. Bolukbasi et al. [8] examined gender associations in word embeddings and how they can influence later language-processing tasks. Caliskan et al. [3] reported that patterns learned from language data can reflect biases present in society.

The issue remains relevant for LLMs because they are trained on large and diverse text collections that may contain historical and demographic biases. These patterns can be learned during training and later appear in model outputs. Blodgett et al. [2] emphasized that fairness in language technology cannot be understood through technical measures alone, since judgments about bias also depend on social and contextual factors. This makes it important to consider how bias is defined, evaluated, and addressed throughout the model lifecycle.

B. Fairness Evaluation Datasets

Evaluating fairness requires benchmarks built to surface specific kinds of bias under controlled conditions. CrowS-Pairs [9] does this by pairing stereotypical and anti-stereotypical sentences across several demographic categories and checking which one a model prefers. WinoBias [10] takes a narrower angle, using occupational stereotypes in coreference resolution to test whether predictions shift with gender.

CrowS-Pairs and WinoBias examine different forms of bias and therefore provide different information about model behavior. Using multiple benchmarks can cover a broader range of demographic contexts and tasks than relying on a single benchmark.

C. Bias Mitigation Approaches

Bias mitigation methods are usually grouped according to when they are applied. Pre-processing methods modify the data before training begins [11]. In-processing methods introduce fairness objectives during model training, often through constrained optimization or adversarial learning [12]. Post-processing methods are applied to model outputs after training. Self-Debias [13] is one example that uses prompting to identify and reduce biased language in generated text.

These methods can improve fairness, but they may also affect predictive performance [4]. Evaluating both fairness and utility helps determine how a mitigation method changes the model's overall behavior.

D. Fairness Frameworks and Governance

Fairness research has also considered broader evaluation and governance frameworks. A review of 121 studies [5] examined bias sources together with methods for detection, evaluation, and mitigation across the NLP lifecycle. The review identified considerable variation in evaluation practices and highlighted the need for more systematic and reproducible procedures.

Fairness and utility have also been studied together in other areas of generative AI. Bochernitsan et al. [6] developed a Pareto-frontier benchmarking framework to examine fairness–utility trade-offs in text-to-image models. Although the application differs from LLMs, the study provides an example of evaluating changes in fairness and utility together. Cookson et al. [7] examined fairness from another perspective by studying how LLMs respond to resource-allocation problems involving fairness considerations.

Broader benchmarking frameworks are also relevant. HELM [14] evaluates language models across multiple scenarios using seven metrics, including fairness, accuracy, robustness, and efficiency. It supports standardized comparisons across models and scenarios. The framework proposed in this study serves a different methodological purpose: rather than benchmarking models across scenarios, it structures the fairness–utility relationship itself into a decision-oriented procedure, incorporating predefined decision criteria and governance-oriented documentation that connect the evaluation to a specific deployment recommendation.

Governance also affects how model evaluations are documented and reviewed. Organizations need records of the evaluation procedure, the evidence obtained, and the basis for decisions about model use. Such documentation can support transparency, accountability, and subsequent auditing [15]. Including these records as part of the evaluation process can also help when deployment decisions are reviewed later.

Three recent studies are particularly relevant to the proposed framework. HALF [16] presents a five-stage pipeline for evaluating fairness in LLMs, with an emphasis on weighting the severity of harms across deployment domains. It does not combine fairness and utility in a dedicated trade-off stage or include governance-oriented reporting. de Lima et al. [17] evaluate fairness and utility within a benchmarking procedure that includes dataset preparation, metric selection, and statistical testing. Their study focuses on adversarial fair-learning models rather than LLMs. Templin et al. [18] propose a five-step audit workflow for LLMs in healthcare. Their approach is specific to healthcare and gives particular attention to stakeholder engagement and model calibration.

These studies share some elements with the proposed framework, but their objectives and scope differ. The framework presented here combines benchmark preparation, fairness assessment, utility evaluation, documentation, and trade-off analysis within one procedure that can be applied across different LLMs and application domains.

E. Research Gap and Framework Contribution

Existing research provides established benchmarks, evaluation measures, and mitigation methods for studying bias in language models. Recent work has also considered fairness–utility relationships and governance requirements, but these elements are often examined separately. For LLMs, methodological guidance remains limited on how to bring them together within a consistent evaluation process. The proposed Fairness–Utility Evaluation Framework addresses this gap by connecting benchmark preparation, model assessment, fairness and utility evaluation, trade-off analysis, and documentation within one procedure. Its structure does not depend on a specific model or benchmark, allowing it to be applied across different LLMs and application contexts.

Table I compares the proposed framework with four closely related studies in terms of their purpose, treatment of fairness and utility, governance-oriented reporting, applicability across models and datasets, and empirical validation.

TABLE I. COMPARISON OF THE PROPOSED FRAMEWORK WITH CLOSELY RELATED WORK

Dimension	[5]	[6]	[7]	[14] (HELM)	Proposed Framework
Primary focus	Review of bias sources, evaluation, and mitigation across the NLP lifecycle	Fairness–utility benchmarking for text-to-image generation	LLM responses to fairness in resource-allocation tasks	Multi-metric LLM benchmarking (fairness, accuracy, robustness, efficiency)	Fairness–utility evaluation framework for LLMs
Joint fairness–utility evaluation	Discussed, not operationalized	Yes (vision domain)	No (fairness-focused study)	Partial (same evaluation suite, no dedicated trade-off stage)	Yes (NLP/LLM domain)
Governance-oriented reporting	Not addressed	Not addressed	Not addressed	No dedicated reporting stage	Dedicated framework stage
Model/dataset independence	N/A (review)	Text-to-image models only	Resource-allocation tasks only	Defined scenarios and metrics	Model- and benchmark-agnostic
Empirical validation	N/A (review)	Yes, multiple T2I models	Yes, multiple LLMs	Yes, 30 models / 42 scenarios	Illustrative application only

TABLE II. MAIN COMPONENTS OF THE PROPOSED FAIRNESS–UTILITY EVALUATION FRAMEWORK

Stage	Inputs	Objective	Output	Key Constraints
Dataset Selection & Preparation	Candidate benchmarks; application context; fairness dimensions of interest	Select and prepare benchmark datasets	Evaluation-ready datasets	Public and established benchmarks are preferred where appropriate; cleaning, validation, and formatting decisions are documented before evaluation.
Model Assessment	Prepared datasets; selected model; prompting and inference configuration	Generate model outputs using standardized inputs	Model predictions and responses	Model version, prompting strategy, and inference parameters — including decoding settings and the random seed where applicable — are specified and recorded before evaluation; repeated runs are used where outputs are stochastic.
Fairness Evaluation	Model outputs; demographic attributes of interest; predefined outcome mapping	Assess fairness-related behavior	Fairness indicators	Outcome mapping and demographic groupings are defined before analysis; fairness metrics are selected according to the fairness question and output structure.
Utility Evaluation	Model outputs; ground-truth labels where available	Measure predictive performance	Utility metrics	Utility metrics must match the benchmark and task design; metrics requiring ground-truth labels are used only when such labels are available.
Trade-off Analysis & Reporting	Fairness indicators; utility metrics; predefined decision criteria	Examine fairness and utility findings together	Evaluation report, deployment recommendation, and evaluation manifest	Thresholds and decision criteria are specified before results are interpreted; any aggregation across metrics, groups, or benchmarks must be predefined and documented; uncertainty or variability is reported where applicable.

III. PROPOSED FAIRNESS–UTILITY EVALUATION FRAMEWORK

A. Overview of the Proposed Framework

The Fairness–Utility Evaluation Framework provides a common process for examining fairness alongside predictive performance. Rather than treating the two as separate parts of model evaluation, it considers fairness measures in relation to how well the model performs on the task under assessment. This approach is consistent with recent calls for more systematic evaluation practices across the LLM lifecycle [5]. In doing so, the framework addresses RQ1 through its five-stage structure, RQ2 through the joint treatment of fairness and utility, and RQ3 through the documentation embedded at each stage.

The process is organized into five stages. It begins with preparing the evaluation data and assessing the model, followed by fairness and utility evaluation. The findings are then considered together in the final stage. Using the same sequence

allows different models and benchmarks to be evaluated through a consistent process across different applications.

The framework is not built around a specific LLM or benchmark. Evaluators can choose the datasets and measures that fit the evaluation at hand while keeping the overall process unchanged. This also means that new models, benchmarks, or metrics can be introduced without having to redesign the framework.

Documentation is maintained as the evaluation progresses. At each stage, evaluators record the relevant inputs, conditions, measures, and decisions. Keeping these details with the evaluation makes it easier to understand how the final assessment was reached and supports the transparency and accountability emphasized in responsible AI governance guidance [15].

The five stages are:

- Dataset Selection and Preparation
- Model Assessment
- Fairness Evaluation
- Utility Evaluation
- Trade-off Analysis and Reporting

Fig. 1 shows how the five stages connect within the overall evaluation process. Table II specifies, for each stage, the required inputs, objective, output, and the key procedural constraints that must be satisfied.



Fig. 1. Overall architecture of the proposed framework.

B. Dataset Selection and Preparation

The first step is to choose benchmark datasets that reflect the fairness dimensions relevant to the application being considered. This choice matters because the data used for evaluation influence which fairness concerns can actually be observed. Where suitable benchmarks are available, the framework gives preference to public and established datasets. Their use supports reproducibility and comparison with findings from other studies [9], [10].

Before evaluation, the selected datasets are prepared to ensure that they can be used consistently. This includes recording the benchmark version, cleaning the data, checking their integrity and consistency, and bringing them into the required format. These preparation steps help prevent data problems from being mistaken for differences in model behavior [5]. Fig. 2 summarizes the process, showing how raw datasets move through cleaning, validation, and formatting before becoming ready for model assessment.



Fig. 2. Dataset preparation pipeline.

C. Model Assessment

After the datasets have been prepared, the selected model is run on standardized inputs taken from the benchmarks. The purpose at this stage is to obtain the model outputs needed for later analysis. Fairness and utility are examined in the stages that follow.

The evaluation procedure should remain consistent across runs, particularly the inputs and prompting method, so that the outputs are generated under comparable conditions, unless prompt variation is itself a deliberate part of the evaluation design, in which case the range of prompts used must be specified in advance. The model version, prompting strategy, inference parameters, and other relevant evaluation conditions are also recorded. These details document how the outputs were produced and allow subsequent assessments to review or reproduce the evaluation setup [15].

Because LLM generation can be stochastic, decoding parameters such as temperature and top-p or top-k, and the random seed where supported by the model or inference framework, must be specified and recorded alongside the model version and prompting strategy. Where outputs vary across repeated runs under the same configuration, the evaluation should include multiple repeated runs, with the number of runs and the resulting variability reported as part of the evaluation record. This variability should subsequently be reflected in the uncertainty measures reported at the Trade-off Analysis and Reporting stage.

D. Fairness Evaluation

$$BS = | P(\hat{y} = 1 \mid A = g_1) - P(\hat{y} = 1 \mid A = g_2) | \quad (1)$$

$$FR = P(\hat{y} = 1 \mid A = g_D) / P(\hat{y} = 1 \mid A = g_A) \quad (2)$$

In Eq. (1), g_1 and g_2 represent the two demographic groups compared under attribute A. A BS value closer to zero indicates a smaller difference between the groups, consistent with the statistical-parity criterion in [20]. Eq. (2) follows the disparate-impact formulation in [21] and compares the positive outcome rate for a disadvantaged group g_D with that of an advantaged group g_A. An FR value close to 1 indicates greater parity, while lower values indicate a larger disparity. The four-fifths rule provides an established reference for interpreting this ratio. BS and FR can be calculated separately for relevant demographic categories and incorporated into the overall fairness assessment; both are established measures in the algorithmic-fairness literature [20], [21] and are used here as components of the proposed framework.

Before BS or FR is calculated, the evaluator must define the outcome variable used in the application. For tasks that already produce categorical predictions, the benchmark label or task decision can be used directly. For open-ended generation, outputs must first be mapped to a predefined categorical or binary outcome using a documented coding rule, classifier, rubric, or human-annotation protocol. The positive outcome must be defined before evaluation and applied consistently across demographic groups. Because this transformation can affect the resulting disparity estimates, the mapping procedure

and its validation or agreement evidence must be reported as part of the evaluation record.

The resulting fairness assessment summarizes the disparities identified during evaluation and provides evidence for determining whether particular demographic groups require further examination.

Fig. 3 shows how model outputs are converted into fairness indicators and incorporated into the assessment report.

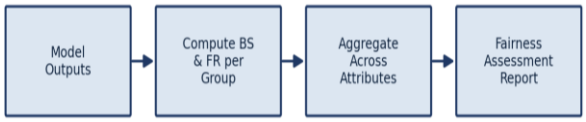


Fig. 3. Fairness evaluation process.

E. Utility Evaluation

The utility evaluation stage examines how well the model performs the task being assessed. Performance is measured using metrics appropriate to the task and benchmark. For prediction tasks with defined ground truth, commonly used measures include Accuracy, Precision, Recall, and F1-score.

Additional measures may be used depending on the task or application. With TP, TN, FP, and FN denoting true positives, true negatives, false positives, and false negatives:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (3)$$

$$Precision = TP / (TP + FP) \quad (4)$$

$$Recall = TP / (TP + FN) \quad (5)$$

$$F1 = 2(P \times R) / (P + R) \quad (6)$$

Accuracy measures the proportion of correct predictions. Precision and Recall describe different aspects of classification performance, while F1 combines the two through their harmonic mean. These metrics can also be reported separately for demographic groups to identify differences in model performance. The results can then be considered alongside the fairness measures defined in Eq. (1) and (2).

The choice of utility metric depends on the benchmark and whether ground-truth labels are available. WinoBias provides coreference-resolution labels that allow measures such as Accuracy and F1 to be calculated. CrowS-Pairs follows a different evaluation design based on model preferences between stereotypical and anti-stereotypical sentence pairs and does not provide conventional ground-truth labels for classification metrics. Utility measures should therefore match the evaluation procedure supported by each benchmark.

Fairness and utility results should be considered together when evaluating a model for potential deployment. Improvements in fairness may affect predictive performance [4], while strong overall performance does not necessarily indicate similar behavior across demographic groups. Considering both provides a clearer view of these differences.

F. Trade-Off Analysis and Reporting

The final stage considers the fairness and utility findings together. The results are compared to examine how the two dimensions relate and whether the overall outcome is suitable for the intended application. This judgment depends on the criteria established for the specific evaluation context. The final report records the benchmarks and metrics used, along with any assumptions or limitations relevant to interpreting the findings. It also documents the evidence supporting the deployment recommendation.

Fairness and utility results are reported at the level of the individual metric, demographic group, and benchmark rather than collapsed into a single overall score, since a single aggregate value can obscure disparities that affect specific groups. Where multiple metrics, groups, or benchmarks are combined into a summary judgment, the aggregation rule must be specified and justified in advance, together with the rationale for any weighting applied. Where estimates are subject to sampling or stochastic variation, the report should include appropriate measures of uncertainty or variability, such as confidence intervals or repeated-run variation, as applicable to the evaluation design. The framework does not mandate a specific statistical test or effect-size measure, as the appropriate procedure depends on the benchmark, sample size, outcome type, and evaluation design.

The final evaluation record is consolidated into an evaluation manifest that identifies the model and benchmark versions, prompting and inference configuration, outcome-mapping rules, fairness and utility metrics, uncertainty information where applicable, and the predefined thresholds, aggregation rules, and decision criteria used in the assessment. The manifest provides a traceable record of the evaluation configuration and resulting deployment recommendation, supporting reproducibility, independent review, and governance oversight.

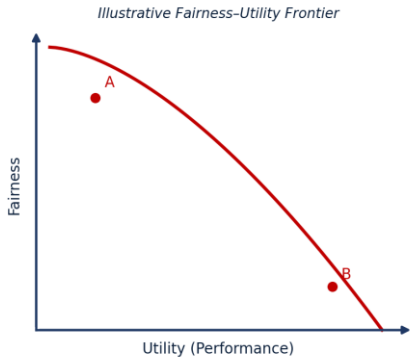


Fig. 4. Fairness–utility trade-off analysis.

This record also serves a governance purpose. Preserving the basis of the assessment can support risk assessment, compliance review, and later auditing [15]. The importance of considering fairness in relation to utility has also been recognized in benchmarking research involving other generative systems [6]. Fig. 4 shows how fairness and utility findings can be interpreted together and identifies cases where the overall outcome may call for further review. The curve represents an illustrative fairness–utility frontier rather than data from an actual evaluation; points

A and B mark two hypothetical outcomes along this frontier, showing how a model could favor fairness (point A) or utility (point B) depending on where it falls relative to the trade-off.

IV. Framework Demonstration: An Illustrative Walkthrough

A. Case Study Overview

This section illustrates how the proposed framework can be used in practice. The example follows its stages in sequence and does not report findings from an empirical experiment.

The example considers a representative LLM evaluation scenario. Its purpose is to show how each stage is applied and how fairness and utility are examined within the same evaluation process.

Following the structure introduced in Section III, the demonstration starts with dataset preparation and model assessment. It then moves to fairness and utility evaluation, followed by trade-off analysis and reporting. This sequence illustrates how evidence from both dimensions can inform the assessment of a model for potential deployment.

For this illustrative scenario, LLaMA-2–7B is designated as the model under consideration [19], and CrowS-Pairs and WinoBias are designated as example benchmark datasets, as they are established resources for examining different forms of bias in language models [9], [10]. Fig. 5 shows how these elements fit into the complete demonstration workflow.



Fig. 5. Illustrative demonstration workflow.

B. Dataset Selection and Preparation

The demonstration begins by selecting benchmarks that match the fairness questions being examined. CrowS-Pairs [9] is used to examine social stereotypes across demographic categories, while WinoBias [10] focuses on gender bias in coreference resolution.

The two datasets address different forms of bias and therefore provide different information about model behavior. In an empirical application, the datasets would be prepared and checked for consistency before model assessment. Any preprocessing or formatting choices would also be documented to support reproducibility. Table III summarizes the two benchmarks.

TABLE III. Example Datasets used in the Demonstration

Dataset	Bias Category	Evaluation Focus
CrowS-Pairs	Social stereotypes	Social stereotype assessment
WinoBias	Gender bias	Gender fairness assessment

C. Model Assessment

Once the datasets are prepared, the model would be run on standardized benchmark inputs. Its outputs would then provide the data needed for the fairness and utility assessments that follow.

LLaMA-2–7B is designated as the reference model for this illustrative demonstration because it is an established open-weight LLM [19]. The model selection is illustrative and does not represent an empirical evaluation of LLaMA-2–7B. The choice of model is specific to the demonstration; the framework itself can be applied to other LLMs using the same procedure.

The evaluation conditions would also need to be documented. These include the model version, prompting procedure, inference parameters, and any other conditions that could affect the generated outputs [15]. Recording these details would allow the evaluation setup to be reproduced in later assessments.

D. Fairness and Utility Assessment

The model outputs would be examined from both fairness and performance perspectives. Keeping the two assessments within the same process allows their findings to be considered together later in the evaluation.

The fairness measures would depend on the application and the demographic attributes being examined. Bias Score and Fairness Ratio are examples of indicators that could be used. Where appropriate, using several measures may reveal disparities that are less apparent when only one indicator is considered [2].

Performance would be evaluated using measures suited to the task and the structure of the benchmark. When ground-truth labels are available, these could include Accuracy, Precision, Recall, and F1-score.

Together, the fairness and performance findings would provide different but related information about the model's behavior. This evidence could help determine whether the model is suitable for the intended application [4]. Table IV summarizes the fairness, utility, and governance dimensions included in the demonstration, along with their corresponding measures and purpose.

TABLE IV. Example Evaluation Dimensions

Dimension	Metrics / Indicators	Purpose
Fairness	Bias Score, Fairness Ratio	Assess fairness-related behavior
Utility	Task-appropriate measures (e.g., Accuracy, Precision, Recall, F1-score where ground truth is available)	Assess predictive performance
Governance	Evaluation documentation and records	Support accountability

E. Trade-Off Analysis and Reporting

For this illustrative scenario, the final stage would consider the fairness and utility findings together to determine whether the model is suitable for the intended application. The assessment may also identify areas where further evaluation or mitigation is needed.

The report would record the benchmarks and measures used, together with relevant assumptions, limitations, and observations. These records would document the basis for the

assessment and provide evidence that can support later review or auditing [15].

Fig. 6 illustrates, in general terms, a decision matrix that could be used to consider fairness and utility together. The zone placement shown is illustrative and does not represent a computed assessment of LLaMA-2–7B. Models with high fairness and high utility would fall within the Preferred Zone, representing the most favorable conditions for deployment. Low results on both dimensions would place a model in the Reject Zone, where substantial improvement would be needed. The Risk Zone represents high utility but low fairness, indicating fairness concerns that require attention. The Review Zone represents high fairness but low utility, indicating a need to improve model performance. The terms High and Low in Fig. 6 are qualitative rather than fixed numerical thresholds. Thresholds for BS, FR, and utility measures should reflect the application, organizational risk tolerance, and relevant regulatory or ethical requirements. These thresholds should be defined in advance by the responsible evaluation or governance team and documented before the results are interpreted. Applying the same predefined thresholds and decision criteria allows independent evaluators to assign models to the four zones on a consistent basis.

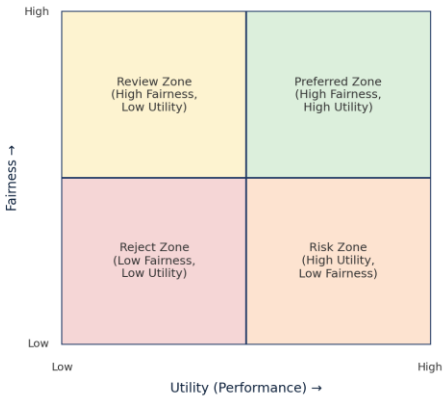


Fig. 6. Illustrative fairness–utility decision matrix for deployment decision support.

F. Discussion

The proposed framework brings these elements into a common sequence, beginning with benchmark preparation and continuing through fairness and utility evaluation to trade-off analysis and reporting. Following the same sequence allows the framework to be applied when the model, benchmark, or application changes.

This differs from approaches that focus on a particular mitigation technique or the evaluation of a specific model. Here, fairness and utility are examined as parts of the same procedure. Because the framework is not tied to a particular LLM architecture, the same sequence can be followed across different models without developing a separate evaluation approach for each one. This also provides a consistent structure for cross-model comparison.

Looking at fairness and utility together matters in practice because an improvement in one does not necessarily leave the other unchanged. Changes in fairness can occur alongside changes in predictive performance [4]. Examining both therefore gives a more complete picture when deciding whether a model is suitable for a particular application.

The framework is also intended to remain usable as LLM evaluation evolves. A different model, fairness measure, or benchmark can be introduced without redesigning the overall process. The individual components can therefore change with the application, available resources, or evaluation requirements while the sequence itself remains stable.

Documentation is part of this process rather than a separate activity at the end. The data and measures used are recorded together with relevant model parameters, assumptions, and limitations. These records allow others to see how the assessment was carried out and to reproduce its main evaluation conditions [15]. They also provide a basis for examining the evidence behind an evaluation when it is later reviewed by researchers or other stakeholders.

Section IV illustrates how this process can be followed using established benchmarks. Its purpose is to show how the stages fit together, including how the fairness and utility assessments relate to one another.

The illustration should not, however, be interpreted as empirical validation of the framework. Assessing its generalizability will require empirical evaluations across multiple LLMs and real deployment contexts. In particular, the use of CrowS-Pairs and WinoBias in Section IV illustrates how the procedure can be applied to established benchmarks; it does not establish external validity across contemporary instruction-following LLMs, nor does it represent the application-specific risks or affected populations relevant to a given deployment context. Its present scope is also limited to fairness and predictive performance. Explainability, robustness, privacy, security, and other responsible AI considerations are not evaluated here, although they could be incorporated into future extensions of the framework.

V. Conclusion

This study introduces a Fairness–Utility Evaluation Framework for assessing large language models. The framework provides a structured way to examine fairness alongside predictive performance while maintaining a record of the evidence generated throughout the evaluation.

Rather than introducing new metrics or benchmarks, the framework brings existing measures, benchmarks, and assessment methods into a common sequence. The same procedure can be applied across different LLMs and application contexts. Because the framework is not tied to a particular model architecture, the model or benchmark can change while the overall evaluation procedure remains consistent.

Section IV illustrates how the framework could be followed using LLaMA-2–7B, CrowS-Pairs, and WinoBias. The example shows how the proposed stages fit together and how fairness and utility findings would be considered within the same assessment. It demonstrates the procedure rather than providing an empirical evaluation and does not report new experimental results.

Documentation is maintained throughout the framework. Key evaluation decisions and supporting evidence are recorded so that the assessment can be traced and reviewed later. These records can support transparency and subsequent review or auditing, particularly when evaluation findings inform decisions about LLM deployment.

The next step in this research program is to test the framework empirically across different LLMs, benchmarks, and application contexts. The framework could also be expanded to address other responsible AI concerns, such as explainability, robustness, privacy, and security. Empirical applications would help assess its generalizability and identify aspects of the procedure that may require refinement. In its current form, the framework provides a methodological basis for examining fairness–utility trade-offs through a documented and reproducible evaluation process.

In relation to the research questions guiding this study, RQ1 is addressed through the five-stage structure presented in Section III; RQ2 through the framework's treatment of fairness and utility within a single evaluation sequence, illustrated in Sections III and IV; and RQ3 through the documentation and governance-oriented reporting embedded at each stage of the framework.

DECLARATIONS

Conflict of Interest: No conflicts of interest are declared.

Funding: This research received no external funding.

Data Availability: The benchmark datasets referenced in this study (CrowS-Pairs and WinoBias) are publicly available from their sources [9], [10]. No new datasets were generated in this study.

Declaration of Generative AI Use: Generative AI tools were used to assist with language editing, formatting, and restructuring during the preparation of this manuscript. The entire manuscript was reviewed and approved by the author. Generative AI was not used as an author or co-author. The research concept, methodology, framework design, analysis, and conclusions are the author's own original work.

REFERENCES

[1] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?," in Proc. 2021 ACM Conf. on Fairness, Accountability, and Transparency, 2021, pp. 610–623.

[2] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "Language (technology) is power: A critical survey of 'bias' in NLP," in Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5454–5476.

[3] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," Science, vol. 356, no. 6334, pp. 183–186, 2017.

[4] E. Sheng, K. W. Chang, P. Natarajan, and N. Peng, "The woman worked as a babysitter: On biases in language generation," in Proc. 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3407–3412.

[5] T. H. Zadid, S. M. Jahin, A. R. Dhruba, S. A. Tanim, A. M. Hamja, M. R. Hasan, M. S. U. Miah, M. I. Islam, and A. A. Mansur, "A review of fairness challenges in natural language processing," Array, vol. 30, art. 100773, 2026.

[6] M. N. Bochernitsan, R. C. Barros, and L. S. Kupssinskü, "A framework for benchmarking fairness–utility trade-offs in text-to-image models via Pareto frontiers," arXiv preprint arXiv:2508.16752, 2025.

[7] B. Cookson, S. Ebadian, and N. Shah, "Fairness perceptions of large language models," in Third IJCAI Workshop on Computational Fair Division (CFD 2025), 2025.

[8] T. Bolukbasi, K. W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? Debiasing word embeddings," Advances in Neural Information Processing Systems, vol. 29, 2016.

[9] N. Nangia, C. Vania, R. Bhalerao, and S. R. Bowman, "CrowS-Pairs: A challenge dataset for measuring social biases in masked language models," in Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 1953–1967.

[10] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K. W. Chang, "Gender bias in coreference resolution: Evaluation and debiasing methods," in Proc. 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2, 2018, pp. 15–20.

[11] R. Zmigrod, S. J. Mielke, R. Cotterell, and I. Augenstein, "Counterfactual data augmentation for mitigating gender stereotypes in language models," in Proc. 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1656–1661.

[12] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in Proc. 2018 AAAI/ACM Conf. on AI, Ethics, and Society, 2018, pp. 335–340.

[13] T. Schick, S. Udupa, and H. Schütze, "Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP," Trans. of the Association for Computational Linguistics, vol. 9, pp. 1408–1424, 2021.

[14] P. Liang, R. Bommasani, T. Lee, et al., "Holistic evaluation of language models," Trans. on Machine Learning Research, 2023.

[15] UNESCO, "Recommendation on the ethics of artificial intelligence," 2021.

[16] A. Mekky, O. El Herraoui, P. Nakov, and Y. Wang, "HALF: Harm-aware LLM fairness evaluation aligned with deployment," arXiv preprint arXiv:2510.12217, 2025.

[17] L. F. F. de Lima, D. R. D. Ricarte, and C. A. Siebra, "A benchmark proposal for non-generative fair adversarial learning strategies using a fairness-utility trade-off metric," Computational Intelligence, vol. 40, no. 5, art. e70003, 2024.

[18] T. Templin, S. Fort, P. Padmanabham, P. Seshadri, R. Rimal, J. Oliva, K. Hassmiller Lich, S. Sylvia, and N. Sinnott-Armstrong, "Framework for bias evaluation in large language models in healthcare settings," npj Digital Medicine, vol. 8, no. 1, art. 414, 2025.

[19] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.

[20] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in Proc. 3rd Innovations in Theoretical Computer Science Conf. (ITCS '12), 2012, pp. 214–226.

[21] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in Proc. 21st ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 2015, pp. 259–268.

Accuracy, Stability, and Computational Efficiency of Deep CNN Architectures for Apple Leaf Disease Classification

Brahim Ouben Hssain[1], Khalil Ladrham[2], Noureddine El Barbri[3], Rachid El Ayachi[4], Abdellah Halimi[5]
Laboratory of Science and Technology for the Engineer (LaSTI),
ENSA, Sultan Moulay Slimane University, Khouribga, Morocco[1, 3]
Intelligent Processing and Security of Systems-Faculty of Sciences,
Mohammed V University, Rabat, Morocco[2]
TIAD Laboratory-Department of Computer Science-Faculty of Sciences and Technology,
Sultan Moulay Slimane University, Morocco[4]
Laboratory of Health Sciences and Technologies-High Institute of Health Sciences,
Hassan 1st University Settat, Morocco[5]

***Abstract*—Reliable apple leaf disease classification requires high predictive performance and transparent computational reporting under heterogeneous field conditions. This study presents an orchard-image benchmark of four ImageNet-initialized convolutional architectures—Xception, EfficientNetB0, DenseNet121, and ResNet50V2—using images acquired at two Moroccan orchard locations (Midelt and Azrou), four cultivars, and three mobile devices. From a curated four-class corpus of 20,000 RGB images, 18,000 images (4,500 per class) formed the five-fold cross-validation benchmark; 2,000 images were isolated before fold construction as an untouched challenge reserve for future independent confirmation and were not evaluated in the present manuscript. All models used a common 224×224 input resolution, a harmonized classification head, Adam optimization, L1/L2 regularization, learning-rate reduction, and validation-loss early stopping. Evaluation combined accuracy, cross-entropy loss, macro-precision, macro-recall, macro-F1, specificity, Cohen's kappa, Matthews correlation coefficient, fold stability, parameter count, model size, GMACs/GFLOPs, and measured CPU training time. All reported values and figures derive from completed HPC-MARWAN runs under the common protocol. Xception achieved the highest observed mean validation accuracy (97.99±0.37%) and aggregate out-of-fold macro-F1 (97.99%), whereas EfficientNetB0 reached 97.33±0.68% mean validation accuracy with 0.78 GFLOPs per image, 5.29 million parameters, and the shortest CPU training time. The results define a descriptive accuracy–cost trade-off within the benchmark; no inferential superiority or device-level deployment latency is claimed.**

Keywords*—Apple leaf disease; convolutional architectures; field conditions; Xception; EfficientNetB0; DenseNet121; ResNet50V2; computational efficiency

I. Introduction

Apple foliar diseases reduce photosynthetic activity, fruit quality, and marketable yield, and field appearance depends on disease stage, cultivar, illumination, leaf orientation, background vegetation, and acquisition device. Shafik et al. [1] enumerated dataset, classification, and deployment problems in plant-disease detection. Pacal et al. [2] highlighted the significance of data quality and evaluation design for the performance of deep learning. Khan et al. [3] built a real-time deep-learning system for apple-specific diagnosis, and Wang et al. [4] proposed an evidence-fusion framework. These studies motivate evaluation under heterogeneous orchard conditions versus controlled imagery alone.

Deep convolutional neural networks (CNNs) learn features hierarchically directly from RGB images, but comparison of published studies is hampered by differences in image resolution, class balance, augmentation, partitioning, validation strategy, and reported metrics. Generalisation can be further affected by field variations in cultivar, illumination, background, and acquisition device. Therefore, a single hold-out split with high accuracy does not in itself validate model stability or transfer to unseen orchard conditions.

Computational reporting is a second limitation. Guan et al. [5] proposed a lightweight plant-disease classifier, Gao et al. [6] developed a lightweight YOLOv8 model for apple leaf disease detection, and Iftikhar et al. [7] integrated an enhanced CNN with a mobile application. These studies demonstrate the relevance of model size and computational demand. However, theoretical FLOPs and CPU training time are computational indicators rather than direct measurements of mobile or edge inference latency.

Information leakage is another methodological concern. Visually related photographs, near-duplicate acquisitions, or augmented derivatives originating from the same source image should not be distributed across training and validation partitions because such overlap can produce optimistic estimates. Shafik et al. [1] and Pacal et al. [2] identify dataset construction, transparent validation, and reproducible partitioning as central methodological requirements.

The contribution is methodological rather than the proposal of another CNN architecture. A common orchard-image and cost-aware benchmark is established for Xception, EfficientNetB0, DenseNet121, and ResNet50V2 using multi-site imagery, leakage-aware five-fold partitioning, a shared classification head, fold-level predictive and stability metrics,

and common computational indicators. Section II reviews related apple and plant-disease studies. Section III defines acquisition, partitioning, preprocessing, training, and evaluation. Section IV reports predictive and computational results; Section V interprets the measured trade-offs; Section VI states limitations and reproducibility requirements; and Section VII summarizes the conclusions and future validation priorities.

II. Related Work

Apple disease classification has progressed from high-capacity CNNs toward attention, segmentation, hybrid, and resource-efficient designs. Yadav et al. proposed AFD-Net for multi-class apple foliar disease recognition and reported strong five-fold performance on the Plant Pathology datasets [8]. Gao et al. introduced BAM-Net for complex-background apple leaf recognition through attention-guided feature extraction [9]. Parashar and Johri combined Inception-v3 with image segmentation and stratified five-fold cross-validation [10]. Si et al. integrated a CNN branch with a Swin Transformer to combine local and long-range representations [11]. Alqahtani et al. coupled EfficientNet with optimization-based feature selection [12], while Ait Nasser and Akhloufi studied a hybrid deep-learning architecture for apple foliar disease detection [13]. AppleLeafNet focused on compact classification [14]; MCDCNet proposed constrained deformable convolutions [15]; Zhu et al. presented a two-stage DeepLabv3+ segmentation framework [16]; LightYOLO-AppleLeafDx focused on lightweight detection [17]; and Zhou et al. explored a multimodal lightweight Transformer [18]. Further context on efficiency is provided by ALNet [19], EnConv [20], edge-device CNN deployment [21], and ensemble hybrid optimisation [22]. The fruit-disease frameworks [23], [24], the crop-disease review by Ngugi et al. [25], the apple-specific review by Kannan et al. [26], and Karasu's optimised apple-scab classifier [27] are other examples of such methodological diversity. Therefore, Table I provides a contextual comparison, rather than a direct performance ranking, across incompatible datasets and protocols.

TABLE I. Methodological Comparison of Representative Apple/Plant Disease Studies

Study	Year	Model/method	Dataset/scope	Validation	Best reported result	Main limitation/distinction
Yadav et al. [8]	2022	AFD-Net (EfficientNet-B3/B4)	Plant Pathology 2020/2021	Five-fold	98.7% accuracy	High-capacity fusion; public datasets
Gao et al. [9]	2023	BAM-Net with attention	Complex-background apple leaves	Hold-out	95.64% accuracy; 95.25% F1	Attention-based; limited cost reporting
Parashar and Johri [10]	2024	Segmentation + Inception-v3	1,821 apple-leaf images; 4 classes	Stratified five-fold	94.76% accuracy	Class imbalance and preprocessing dependence
Si et al. [11]	2024	CNN + Swin Transformer	Apple disease images; 4 classes	Hold-out	97.32% accuracy	Hybrid local/global feature modelling
Alqahtani et al. [12]	2023	EfficientNet + Sailfish optimization	9,714 apple-leaf images; 4 classes	Train/test	99.39% comparison result	Optimization-assisted high-complexity pipeline
Ali et al. [14]	2024	AppleLeafNet	Apple leaf disease images	Train/validation/test	98.25% condition identification	Lightweight deployment-oriented design
Zou et al. [17]	2025	LightYOLO-AppleLeafDx	Apple leaf disease detection	Detection benchmark	96.5% mAP@0.5	5.2 MB model; real-time orientation
Adithiyaa et al. [19]	2026	ALNet	Apple/grape/cherry leaf datasets	Five-fold + hold-out	99.95% apple accuracy	0.17 M-parameter lightweight model
Thanjaivadivel et al. [20]	2025	EnConv	39 plant-disease classes	Comparative classification	99.87% accuracy	Cross-crop enhanced CNN
Present benchmark	2026	Xception / EfficientNetB0 / DenseNet121 / ResNet50V2	Multi-site orchard corpus; 18,000 benchmark images	Leakage-aware five-fold	97.99% mean accuracy (Xception)	Joint accuracy-stability-cost analysis

The comparison highlights substantial variation in dataset origin, validation design, model complexity, and deployment evidence. The present work differs by treating predictive accuracy, fold stability, computational complexity, and runtime as joint selection criteria under one common orchard-image benchmark.

III. Materials and Methods

A. Study Design and Field Acquisition

Images were collected in commercial apple orchards in Midelt and Azrou, Morocco. The two sites differ in altitude, local microclimate, orchard conditions, illumination, and background context. Photographs were taken outdoors in natural light with leaves still attached to the trees and at morning, midday, and afternoon acquisition times to maintain realistic illumination variability.

Four cultivars were represented in the field records: Gala, Golden Delicious, Starking, and Red Delicious. Three mobile devices were used: Samsung Galaxy S21 Ultra, Xiaomi Redmi 13 Pro, and iPhone 12 Pro Max. The analytical labels were image-level visual symptom categories: powdery mildew, healthy apple leaf, apple scab, and apple mosaic/chlorotic symptoms. The available benchmark records do not document laboratory confirmation or formal multi-expert adjudication for every image; consequently, the labels are interpreted as visual symptom categories rather than etiologically confirmed diagnoses. This limitation is particularly relevant to the mosaic/chlorotic category, which may have multiple causal mechanisms. Label errors identified during quality screening were removed; however, a separate adjudication record for ambiguous cases is not available for the present benchmark. Representative examples are shown in Fig. 1. Phytotoxicity was excluded because it represents abiotic or chemical stress rather than a single pathogen-defined category.

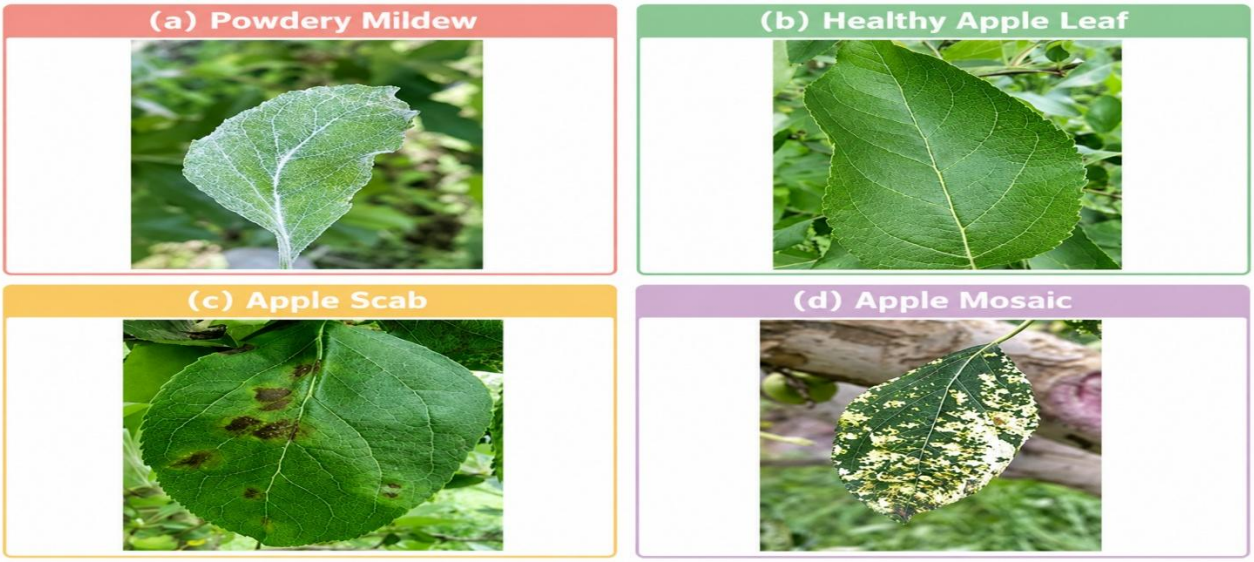


Fig. 1. Representative examples of the four analytical classes: powdery mildew, healthy apple leaf, apple scab, and apple mosaic/chlorotic symptoms.

B. Benchmark Construction and Leakage Control

The curated corpus contained 20,000 RGB images, corresponding to 5,000 images per class after controlled balancing and quality screening. A total of 18,000 images (4,500 per class) were assigned to five-fold cross-validation. The remaining 2,000 images were isolated before fold construction as an untouched challenge reserve. The reserve was excluded from architecture selection, early stopping, hyperparameter adjustment, fold construction, and all reported metrics. Its evaluation is deferred entirely to future independent confirmation, and no reserve-set performance is claimed in the present manuscript.

Before fold assignment, corrupted files, severe blur, label errors, exact duplicates, and visually redundant derivatives were screened. The guaranteed grouping unit was the original source image together with its augmented derivatives; all derivatives from a source image remained in the same fold. Complete leaf- and tree-level identifiers were not available for all acquisitions, so strict leaf- or tree-level independence across folds cannot be guaranteed. Where capture-session identifiers were available, near-duplicate images were grouped to reduce acquisition-context leakage; orchard-level separation was not imposed. Each iteration used 14,400 training images and 3,600 validation images, and every benchmark image served as validation data once. The reported five-fold results should therefore be interpreted as leakage-aware source-image-level internal validation rather than leave-one-tree, leave-one-session, or leave-one-orchard external validation. Fig. 2 summarizes the partitioning sequence.

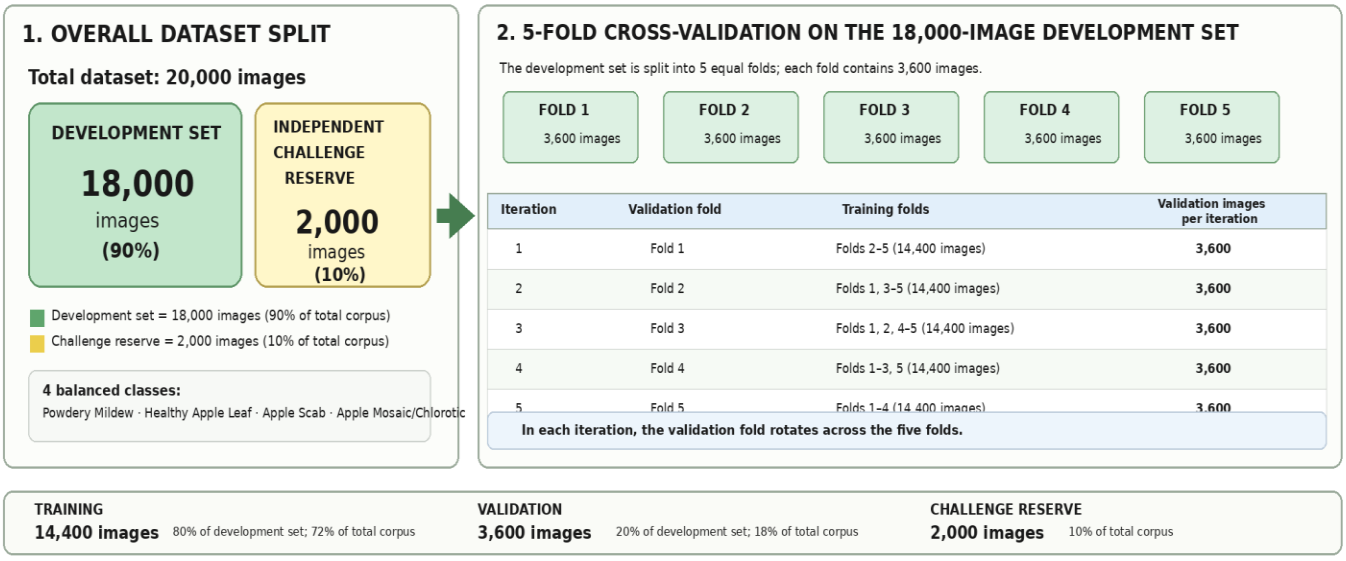


Fig. 2. Orchard-image benchmark design. The 2,000-image challenge reserve is isolated from the five-fold benchmark and is not used to derive the reported cross-validation results.

C. Preprocessing and Augmentation

Images were decoded in RGB format and resized to 224×224×3 pixels. Aspect-ratio-preserving padding was applied when necessary to reduce geometric distortion. Architecture-specific input normalization followed the preprocessing conventions of the corresponding Keras backbones. Validation images underwent deterministic resizing and normalization only.

Mild online augmentation was restricted to training folds. The transformation set included horizontal flipping, rotations within ±15°, translations up to ±8%, zoom up to ±10%, and moderate brightness/contrast adjustment. The limits were selected to preserve disease morphology while increasing robustness to leaf orientation and illumination. Augmentation was performed after fold assignment so that transformed variants could not cross the training-validation boundary.

D. CNN Architectures and Shared Classification Head

The four backbones were selected to represent established image-classification design families under one common protocol: Xception uses depthwise separable convolutions, EfficientNetB0 uses compound scaling, DenseNet121 uses dense feature reuse, and ResNet50V2 uses pre-activation residual learning. Shafik et al. [1] and Pacal et al. [2] provide broad plant-disease context; ALNet [19] and EnConv [20] illustrate recent lightweight and cross-crop alternatives; Ngugi et al. [25] review crop-disease diagnosis; Kannan et al. [26] summarise apple-leaf advances. The chosen set is representative, not exhaustive, and is not claimed to include the strongest current architecture. LightYOLO-AppleLeafDx was retained as a literature comparator, but not inserted into the controlled classification benchmark, as detection-specific or materially different pipelines.

For each backbone, the original classification layer was removed and replaced by global average pooling, batch normalization, a 256-unit ReLU dense layer, dropout, and a four-unit Softmax output. L1 and L2 penalties were applied to the dense layer, while pretrained batch-normalization layers remained in inference mode during fine-tuning. Table II reports architecture-specific batch sizes, dropout values, learning rates, fine-tuned regions, and convergence limits. These settings were selected to obtain stable training for each backbone; consequently, the comparison represents performance under defined architecture-specific configurations rather than an ablation that isolates backbone architecture as the sole causal factor.

TABLE II. ARCHITECTURE-SPECIFIC TRAINING CONFIGURATION

Model	Batch	Dropout	Head LR	Fine-tune LR	Fine-tuned region	Max. epochs	Observed stop range
Xception	8	0.40	1×10^{-4}	3×10^{-6}	Exit flow + upper middle-flow blocks	50	30–35
EfficientNetB0	24	0.30	3×10^{-4}	1×10^{-5}	Last 30–40 layers	50	40–47
DenseNet121	12	0.35	2×10^{-4}	5×10^{-6}	Final dense block	50	42–48
ResNet50V2	16	0.35	2×10^{-4}	5×10^{-6}	Final residual stage	50	40–48

E. Training and Convergence Control

Training and all recorded fold outputs used in Sections IV–V were generated with TensorFlow/Keras on HPC-MARWAN using eight CPU cores and 16 GB RAM. Adam used $\beta1 = 0.9$, $\beta2 = 0.999$, and $\varepsilon = 10^{-7}$. Each fold started from the same ImageNet initialization instead of inheriting weights from a previous fold. The classification head was optimized first, followed by low-learning-rate fine-tuning of an architecture-specific upper region. A single training run was conducted for each fold; repeated runs across multiple random seeds were not performed, so stochastic variability beyond fold-to-fold variation is not estimated.

A maximum of 50 epochs was initially specified for every architecture to provide sufficient optimization capacity. Validation loss was monitored with early stopping (patience = 5 epochs; minimum improvement = 10^{-4}), and the best validation-loss weights were restored. ReduceLROnPlateau used a reduction factor of 0.2, patience of two epochs, a one-epoch cooldown, and a minimum learning rate of 10^{-7}.

For Xception, convergence occurred earlier than the maximum training limit: the five folds stopped at epochs 30, 33, 31, 35, and 34. The training process was therefore not arbitrarily restricted to 35 epochs. Instead, the early-stopping criterion terminated optimization when validation accuracy and validation loss had reached a stable plateau over the predefined patience interval. Continuing beyond this point would have increased computational cost without providing a meaningful validation improvement, while potentially increasing overfitting risk.

F. Evaluation Metrics and Computational Analysis

Out-of-fold predictions were used to compute accuracy, macro-precision, macro-recall, macro-F1, macro-specificity, Cohen's kappa, and the Matthews correlation coefficient (MCC). For class c, TPc, TNc, FPc, and FNc denote the standard confusion-matrix quantities. Macro-averaging gives equal weight to each class:

$$\text{Accuracy} = \Sigma c\ TPc / N, \quad (1)$$

$$\text{Pmacro} = (1/C)\ \Sigma c\ [TPc /(TPc + FPc)], \quad (2)$$

$$\text{Rmacro} = (1/C)\ \Sigma c\ [TPc /(TPc + FNc)], \quad (3)$$

$$\text{F1macro} = (1/C)\ \Sigma c\ [2PcRc /(Pc + Rc)] \quad (4)$$

Categorical cross-entropy was used as the data loss. Fold stability was summarized by the standard deviation and coefficient of variation (CV) of validation accuracy, and the mean train-validation accuracy gap was reported descriptively. Because the five folds share overlapping training observations, they are not independent statistical samples. Complete leaf- and

tree-level group identifiers were also unavailable for all images; therefore, independent-sample significance tests and group-level resampling confidence intervals were not reported. In particular, the 0.66-percentage-point difference between Xception and EfficientNetB0 is treated as a descriptive benchmark difference and not as evidence of statistical superiority.

Computational cost was characterized by parameter count, model size, multiply-accumulate operations (GMACs), forward-pass GFLOPs, and measured wall-clock CPU training time per epoch. A convention of one MAC ≈ two FLOPs was applied consistently. Validation-accuracy points per GFLOP, per million parameters, and per CPU training minute were calculated only as descriptive within-benchmark ratios. CPU training time was measured on HPC-MARWAN; derived throughput and five-fold burden are arithmetic summaries. These values do not represent mobile/edge inference latency, energy, or peak memory.

IV. Results

A. Fold-Wise Predictive Performance

Predictive results and fold-specific stopping epochs are provided in Table III. The original four-panel learning-curve format is preserved: mean training accuracy, mean validation accuracy, mean training loss, and mean validation loss are shown separately to allow direct assessment of convergence and train-validation agreement. For visualisation, folds with different early-stopping points are aligned to a normalised epoch-progress scale, which is only relevant for the graphic display and not the original stopping epochs or reported metrics. The curves demonstrate progressive saturation of accuracy and reduction of loss without significant late-stage divergence. Xception showed the best validation trajectory among the tested configurations, while the other architectures converged at slightly lower levels. The four trajectories are shown in Fig. 3, and the fold-specific stopping epochs are listed in Table III.

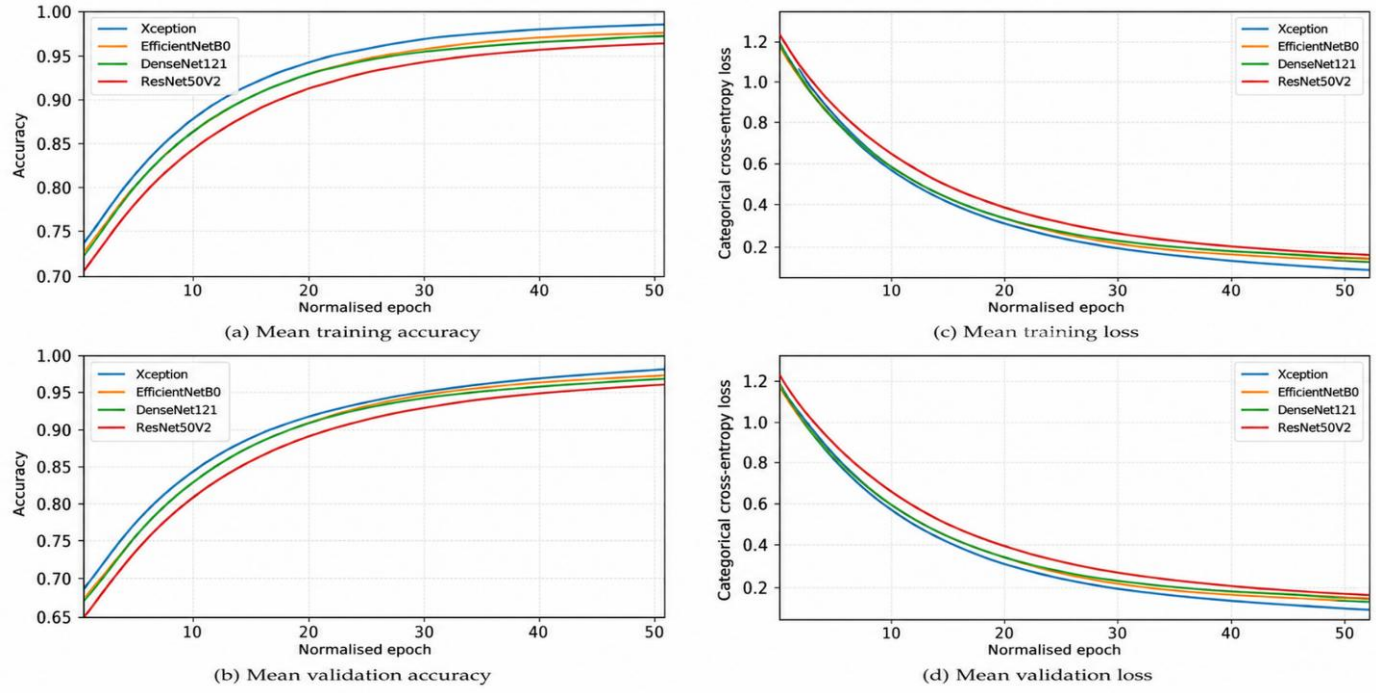


Fig. 3. Mean five-fold learning trajectories shown on a normalized epoch-progress scale: (a) training accuracy, (b) validation accuracy, (c) training loss, and (d) validation loss. The actual fold-specific early-stopping epochs are reported in Table III.

TABLE III. Fold-Wise Results for the Four CNN Architectures

Model	Fold	Stopping epoch	Train Acc. (%)	Val. Acc. (%)	Train Loss	Val. Loss
Xception	1	30	98.30	97.50	0.091	0.128
	2	33	98.58	97.78	0.076	0.107
	3	31	98.70	98.00	0.066	0.091
	4	35	98.92	98.44	0.051	0.066
	5	34	98.80	98.22	0.059	0.078
EfficientNetB0	1	40	97.15	96.50	0.158	0.188
	2	45	97.50	96.86	0.134	0.159
	3	43	97.82	97.31	0.114	0.136

Model	Fold	Stopping epoch	Train Acc. (%)	Val. Acc. (%)	Train Loss	Val. Loss
	4	47	98.50	98.19	0.081	0.083
	5	42	98.02	97.78	0.096	0.107
DenseNet121	1	44	96.78	96.11	0.171	0.198
	2	42	97.05	96.47	0.151	0.177
	3	46	97.40	96.83	0.132	0.154
	4	48	97.92	97.50	0.087	0.096
	5	43	97.62	97.17	0.108	0.124
ResNet50V2	1	46	96.20	95.50	0.184	0.218
	2	40	96.36	95.75	0.165	0.194
	3	44	96.55	96.00	0.145	0.171
	4	48	96.89	96.50	0.099	0.112
	5	42	96.72	96.25	0.121	0.139

B. Mean Performance, Agreement and Stability

Table IV summarizes fold-level performance, agreement, convergence, and stability. Xception achieved the highest mean validation accuracy at 97.99±0.37%, followed by EfficientNetB0 at 97.33±0.68%. DenseNet121 reached 96.82±0.55%, whereas ResNet50V2 reached 96.00±0.40%. Aggregate out-of-fold macro-F1 values were 97.99%, 97.33%, 96.82%, and 96.00%, respectively. The same ranking was observed for validation loss, kappa, and MCC.

The coefficient of variation of validation accuracy was 0.37% for Xception, 0.70% for EfficientNetB0, 0.57% for DenseNet121, and 0.41% for ResNet50V2. The corresponding mean train-validation accuracy gaps were 0.67, 0.47, 0.54, and 0.54 percentage points. These descriptive values show low fold dispersion for Xception and slightly greater dispersion for EfficientNetB0; no inferential significance is assigned to differences between architectures.

Aggregate out-of-fold metrics recomputed from the 18,000-prediction confusion matrices are reported in Table V. Because the benchmark is class-balanced, balanced accuracy equals macro-recall and is numerically very close to overall accuracy.

TABLE IV. MEAN PREDICTIVE AND STABILITY METRICS

Model	Train Acc. (%)	Val. Acc. (%)	Val. Loss	Kappa	MCC	CV Acc. (%)	Mean gap (pp)	Stop epoch
Xception	98.66±0.24	97.99±0.37	0.094±0.024	0.9732	0.9732	0.37	0.67	32.6±2.1
EfficientNetB0	97.80±0.51	97.33±0.68	0.135±0.041	0.9644	0.9644	0.70	0.47	43.4±2.7
DenseNet121	97.35±0.45	96.82±0.55	0.150±0.041	0.9576	0.9576	0.57	0.54	44.6±2.4
ResNet50V2	96.54±0.28	96.00±0.40	0.167±0.042	0.9467	0.9467	0.41	0.54	44.0±3.2

TABLE V. AGGREGATE OUT-OF-FOLD CLASSIFICATION METRICS

Model	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Balanced Acc. (%)	Macro-Specificity (%)
Xception	97.99	97.99	97.99	97.99	99.33
EfficientNetB0	97.33	97.33	97.33	97.33	99.11
DenseNet121	96.82	96.82	96.82	96.82	98.94
ResNet50V2	96.00	96.00	96.00	96.00	98.67

C. Class-Level Error Analysis

Before examining the confusion matrices, the aggregate metrics indicate that the principal remaining errors are class-specific rather than a consequence of unstable global convergence. The combined out-of-fold matrices allow these errors to be localized by true and predicted class. Xception correctly classified 4,418 powdery mildew images, 4,408 healthy images, 4,394 apple scab images, and 4,418 mosaic/chlorotic-symptom images. Its errors remain concentrated in a small number of scab-versus-healthy or scab-versus-chlorotic cases. The corresponding matrices for all four models are compared in Fig. 4.

For Xception, class-specific F1-scores derived from the combined matrix were 98.30% for powdery mildew, 97.87% for healthy leaves, 97.67% for apple scab, and 98.12% for

mosaic/chlorotic symptoms. Apple scab therefore remained the lowest-F1 class despite the strong overall performance.

EfficientNetB0 retained strong diagonal concentration but produced more off-diagonal errors around apple scab than Xception. DenseNet121 showed a further increase in scab and mosaic/chlorotic errors, and ResNet50V2 produced the broadest off-diagonal distribution. The confusion matrices establish where errors occurred but do not identify their visual cause; any attribution to shared necrotic, chlorotic, or textural appearance remains a hypothesis that requires lesion-level explanation or expert review.

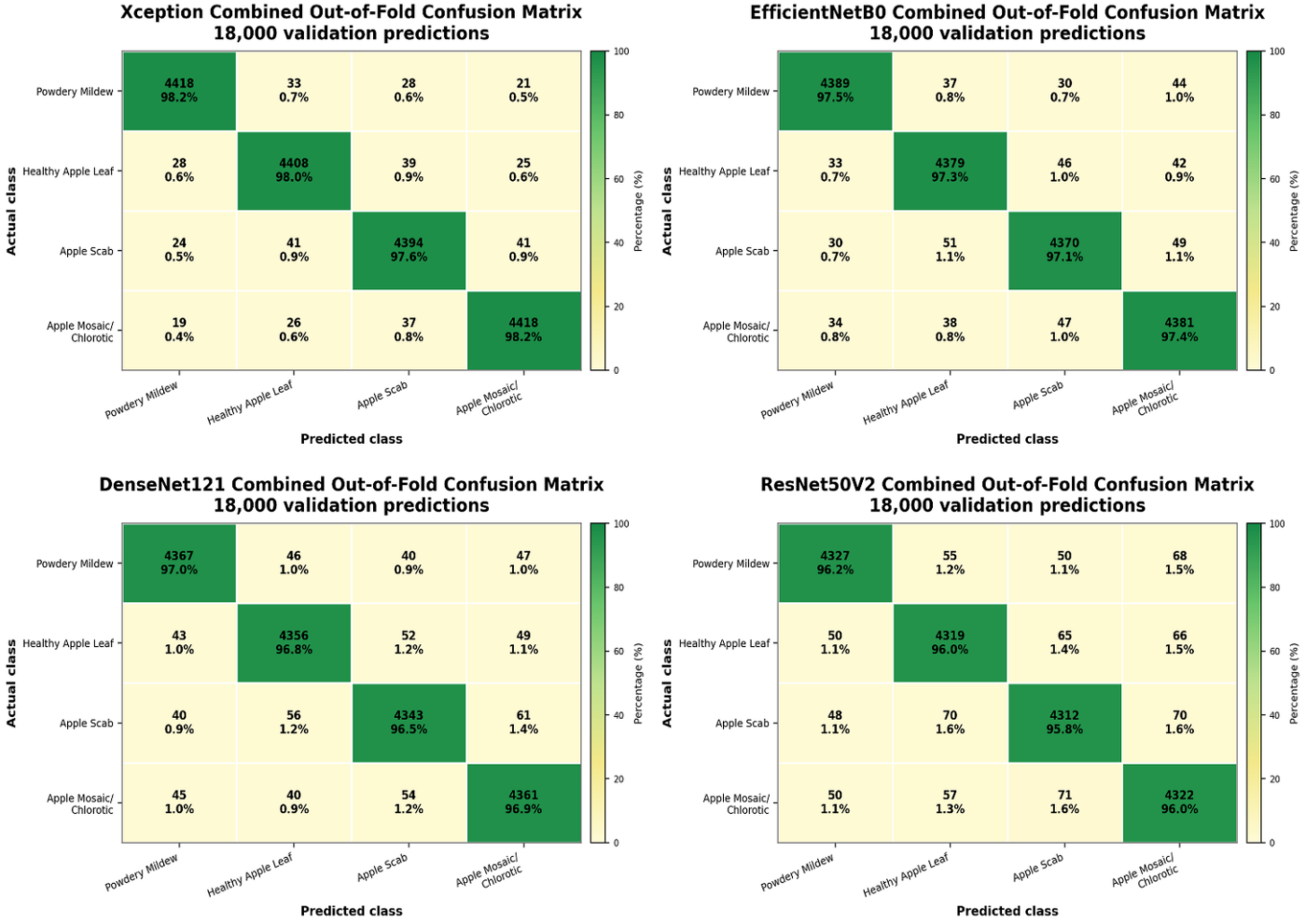


Fig. 4. Combined out-of-fold confusion matrices: (a) Xception, (b) EfficientNetB0, (c) DenseNet121, and (d) ResNet50V2. Each matrix contains 18,000 validation predictions.

D. Computational Cost and Accuracy-Efficiency Frontier

EfficientNetB0 has 5.29 million parameters, 0.78 GFLOPs per image, and a mean measured CPU training time of 35.42\pm0.62 min per epoch.

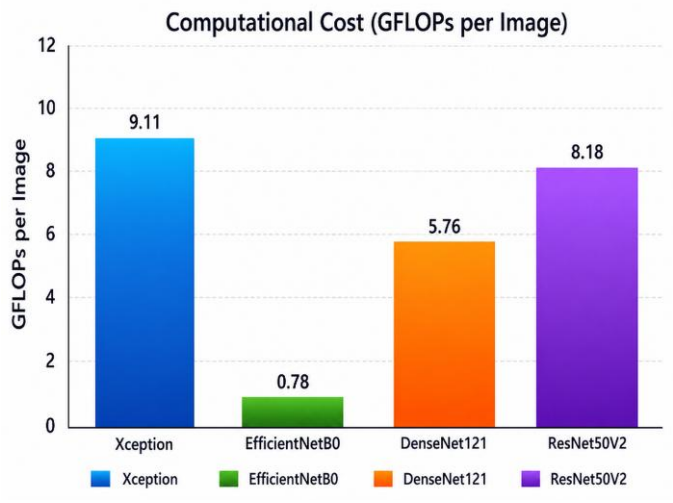
Xception achieved the highest mean validation accuracy and the highest CPU training time (90.28±1.38 min/epoch) with 21.40 million parameters and 9.11 GFLOPs/image. For DenseNet121, we used an intermediate cost position with 7.31 million parameters, 5.76 GFLOPs, and 65.06 min/epoch. ResNet50V2 had the largest parameter footprint (24.10 million (91.9 MiB)) and the lowest mean validation accuracy.

The reported mean per-epoch training times correspond to computed training throughputs of approximately 2.66, 6.78, 3.69 and 3.34 images/s for Xception, EfficientNetB0, DenseNet121 and ResNet50V2 on 14,400 training images per fold. Combining stopping epochs with mean per-epoch times produces estimated aggregate five-fold CPU training burdens of 245.3, 128.1, 241.8 and 263.8 h, respectively. These are arithmetic estimates from training measurements, and should not be interpreted as inference throughput on deployment hardware.

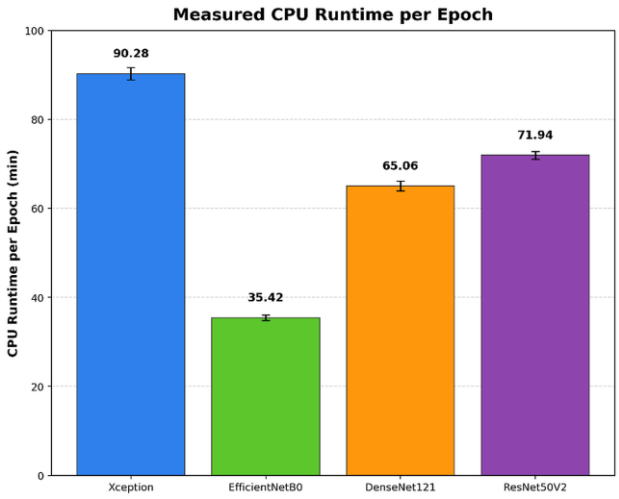
The predicted and measured rankings of resources were not the same. Xception had the highest validation accuracy, but EfficientNetB0 was 0.66 percentage points behind, with substantially fewer GFLOPs, parameters, and CPU training minutes per epoch. From this descriptive accuracy-cost perspective, DenseNet121 and ResNet50V2 were less well placed. These measured trade-offs are visualised in Fig. 5 without implying device-level deployment performance.

TABLE VI. COMPUTATIONAL COST AND RESOURCE-NORMALIZED PERFORMANCE

Model	Params (M)	GMACs	GFLOPs	Runtime (min/epoch)	Size (MiB)	Acc./GFLOP	Acc./M-param	Acc./min
Xception	21.40	4.56	9.11	90.28±1.38	81.6	10.76	4.58	1.09
EfficientNetB0	5.29	0.39	0.78	35.42±0.62	20.2	124.78	18.40	2.75
DenseNet121	7.31	2.88	5.76	65.06±1.11	27.9	16.81	13.24	1.49
ResNet50V2	24.10	4.09	8.18	71.94±0.86	91.9	11.74	3.98	1.33



(a) Approximate forward-pass complexity.



(b) CPU runtime per epoch.

Fig. 5. Computational comparison of the four architectures: (a) profiled forward-pass GFLOPs per image and (b) measured wall-clock CPU runtime per epoch. EfficientNetB0 provides the lowest computational requirement, whereas Xception provides the highest observed mean validation accuracy.

E. Additional Graphical Synthesis

The validation accuracy, MCC, fold stability, accuracy per GFLOP, accuracy per CPU training minute and compactness for the four architectures are summarised in Fig. 6 after min-max normalisation. Before normalisation, lower-is-better quantities were inverted. The plot is descriptive and compares measured accuracy-cost profiles but does not establish causal effects of individual architectural mechanisms.

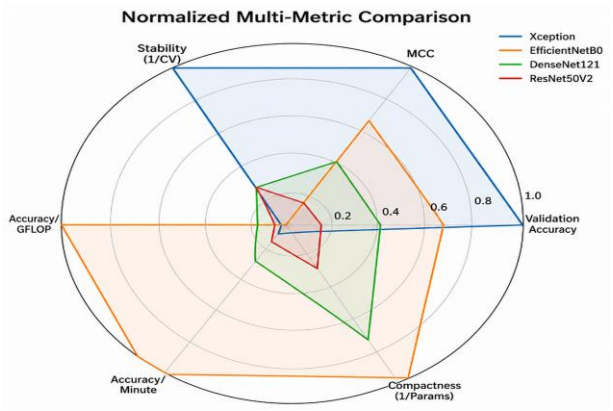


Fig. 6. Normalized multi-metric comparison of the four architectures across predictive quality, stability, and computational efficiency.

Fig. 7 shows complementary synthesis plots of mean validation accuracy vs measured CPU training time per epoch. Bubble size indicates parameter count and labels report GFLOPs. We observe EfficientNetB0 in the lowest-cost region of the four configurations evaluated, and Xception with the highest mean validation accuracy at a higher measured training cost.

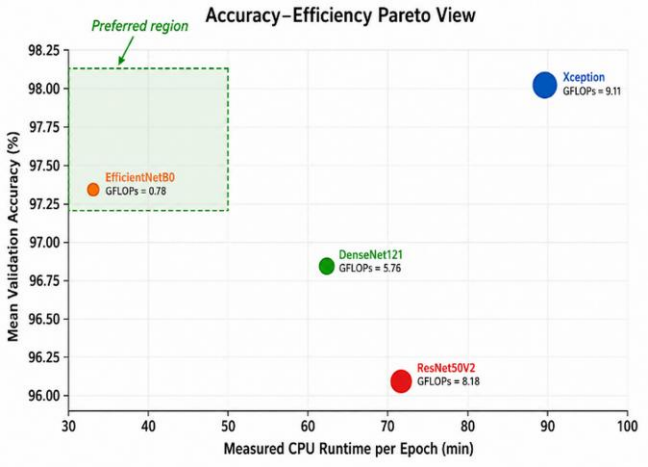


Fig. 7. Accuracy-efficiency Pareto view using mean validation accuracy, measured CPU runtime per epoch, parameter footprint, and GFLOPs.

Fig. 8 reports accuracy per GFLOP, accuracy per million parameters, and accuracy per CPU training minute. These ratios identify EfficientNetB0 as the lowest-cost configuration under the reported benchmark indicators, but they do not establish mobile or edge inference efficiency because device-level latency and energy were not measured.

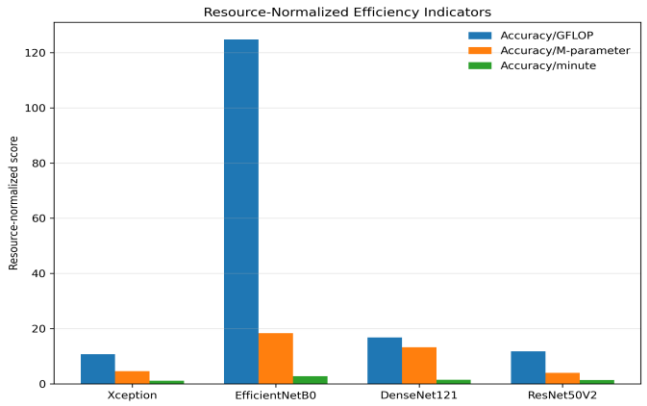


Fig. 8. Resource-normalized efficiency indicators derived from Table VI: accuracy per GFLOP, accuracy per million parameters, and accuracy per minute.

Fig. 9 summarizes the fold-wise distributions of validation accuracy and validation loss. Xception shows the smallest accuracy spread among the two highest-performing models, while EfficientNetB0 shows the largest fold-to-fold accuracy spread. The boxplots are descriptive because the folds are not independent statistical samples.

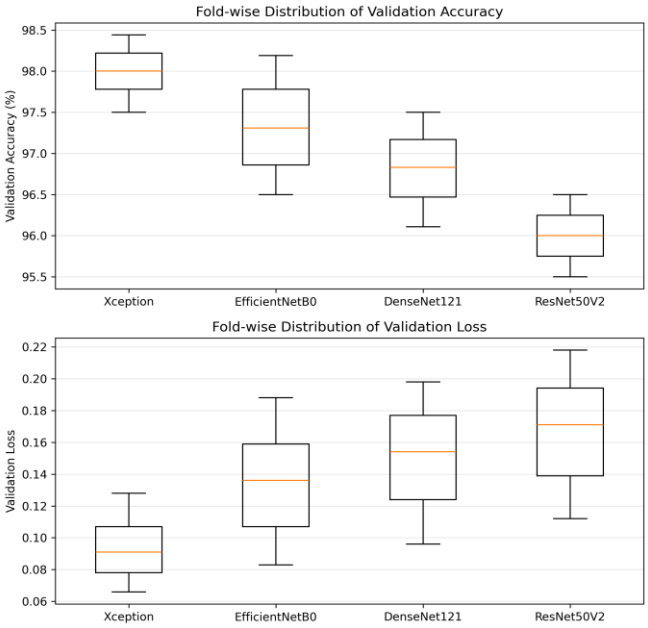


Fig. 9. Fold-wise distribution of validation accuracy and validation loss across the five cross-validation folds.

V. Discussion

Architecture selection for orchard-image classification is a multi-objective decision rather than a ranking by accuracy alone. Xception recorded the highest mean validation accuracy, aggregate out-of-fold macro-F1, kappa, and MCC, together with the lowest mean validation loss. The experiment does not isolate the architectural mechanism responsible for these differences, and no causal attribution is made to depthwise separable convolution or another individual design component. The measured cost of the Xception configuration was substantially greater than that of EfficientNetB0.

EfficientNetB0 was 0.66 percentage points below Xception in mean validation accuracy while using 91.4% fewer GFLOPs, 75.3% fewer parameters, and 60.8% less mean CPU training time per epoch. These measurements indicate a lower computational footprint under the benchmark protocol. The accuracy difference is descriptive; no claim of inferential statistical significance is made, and the training-time measurements do not establish mobile or edge inference latency, throughput, memory, or energy behavior.

DenseNet121 occupied an intermediate accuracy–cost position, whereas ResNet50V2 had the largest parameter footprint and the lowest mean validation accuracy under the specified configuration. These outcomes are configuration-specific and do not establish intrinsic superiority or weakness of an architecture family because optimization settings and training stochasticity were not isolated through a dedicated ablation or repeated-seed design.

Apple scab had the lowest class-specific F1-score for Xception. The confusion matrices establish this empirical error pattern but do not determine its cause. Visual overlap with chlorotic or necrotic symptoms is therefore treated only as a hypothesis. Natarajan et al. [28] used explainable-AI visualizations for plant-disease diagnosis, while Alhammad et al. [29] applied explainable deep learning to potato leaf disease classification. Comparable lesion-level analysis is deferred to future work before any biological interpretation of the scab confusion pattern is asserted.

Direct numerical comparison with prior studies requires caution because datasets, augmentation, acquisition conditions, class definitions, and validation procedures differ. Yadav et al. [8] reported five-fold AFD-Net results; AppleLeafNet emphasized compact apple-disease recognition [14]; ALNet reported a lightweight design [19]; and EnConv addressed multi-crop leaf-disease classification [20]. The four backbones in the present benchmark were selected as representative established families rather than as an exhaustive state-of-the-art set. LightYOLO-AppleLeafDx and other task-specific lightweight models were not reimplemented; therefore, no claim is made that the evaluated four models constitute the strongest available operating points.

A field benchmark should distinguish predictive quality, fold stability, computational complexity, and hardware-specific deployment evidence. The present study directly reports the first three dimensions together with CPU training time and forward-pass complexity, but it does not measure device-level inference latency, throughput, energy consumption, or peak memory. Consequently, the conclusions are limited to the measured benchmark accuracy–cost trade-off rather than verified mobile or edge deployment suitability.

VI. Limitations and Reproducibility

The benchmark spans two Moroccan orchard locations (Midelt and Azrou), four cultivars, and three mobile devices, but it does not cover all regions, seasons, disease stages, or acquisition conditions. The guaranteed independence unit is the original source image and its augmentation lineage; complete leaf- and tree-level identifiers are not available for all acquisitions, so stricter biological independence across folds

cannot be confirmed. Labels are treated as visual symptom categories rather than universally laboratory-confirmed etiologies, and the mosaic/chlorotic category may contain heterogeneous causal mechanisms. The reported cross-validation results therefore quantify internal source-image-level benchmark stability, not external orchard-level or etiological generalization.

The 2,000-image challenge reserve was isolated before cross-validation and excluded from model selection and all headline metrics. It is intentionally not evaluated in this manuscript; reserve-set confirmation is deferred entirely to future work to preserve its independent role. The reported results constitute the complete five-fold analysis of the 18,000-image development benchmark. Future confirmation should additionally use leave-one-orchard-out, leave-one-device-out, or leave-one-season-out protocols where the corresponding group identifiers are available.

Architecture-specific optimization settings and a single run per fold limit causal interpretation of backbone differences because random-seed variability was not separated from optimization effects. Group-level confidence intervals were not calculated because complete leaf- and tree-level identifiers were unavailable; an image-level bootstrap would overstate independence. Future studies should retain leaf, tree, session, device, and orchard identifiers, repeat training across random seeds, and apply group-level resampling or permutation procedures. Reproducibility also requires preserved fold assignments, augmentation lineage, software versions, training logs, checkpoints, profiler outputs, and out-of-fold probabilities under identical hardware and data-loader settings.

VII. Conclusion and Future Work

An orchard-image five-fold benchmark compared Xception, EfficientNetB0, DenseNet121, and ResNet50V2 for four-class apple leaf disease classification. Xception achieved the highest mean validation accuracy (97.99±0.37%) and aggregate out-of-fold macro-F1 (97.99%), while EfficientNetB0 reached 97.33±0.68% with markedly fewer parameters, lower GFLOPs, and shorter CPU training time. DenseNet121 occupied an intermediate position, and ResNet50V2 the least favorable measured accuracy–cost position. Because no inferential architecture test or device-level inference benchmark was conducted, these findings identify descriptive benchmark operating points rather than statistically or deployment-validated superiority.

Mean accuracy alone is insufficient for model selection. Fold dispersion, train-validation gap, agreement metrics, parameter count, FLOPs, training time, and memory footprint provide complementary evidence. The early-stopping protocol also shows that training duration should follow convergence rather than an arbitrary epoch count; Xception stopped after 30–35 epochs despite a maximum limit of 50 epochs.

Future work will prioritize evaluation of the untouched challenge reserve and external validation across unseen orchards, devices, cultivars, and seasons. Complete leaf/tree/session identifiers will support strict group-aware partitioning, repeated random-seed runs, and group-level uncertainty estimates. Explainable-AI analysis will also be used for lesion localization and class-specific failure assessment before causal interpretations of error patterns are made.

We will explore structured pruning, quantisation, and knowledge distillation to reduce model size and inference cost. Before making claims about mobile or edge suitability, hardware-specific latency, throughput, peak memory, and energy measurements will be gathered. The selected efficiency-oriented configuration can be embedded in an orchard diagnostic interface, with confidence reporting and class-specific explanations.

References

[1] W. Shafik, A. Tufail, A. Namoun, L. C. de Silva, and R. A. A. H. M. Apong, “A systematic literature review on plant disease detection: Motivations, classification techniques, datasets, challenges, and future trends,” IEEE Access, vol. 11, pp. 59174–59203, 2023, doi: 10.1109/ACCESS.2023.3284760.

[2] I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, and R. Martinek, “A systematic review of deep learning techniques for plant diseases,” Artificial Intelligence Review, vol. 57, art. 304, 2024, doi: 10.1007/s10462-024-10944-7.

[3] A. I. Khan, S. M. K. Quadri, S. Banday, and J. L. Shah, “Deep diagnosis: A real-time apple leaf disease detection system based on deep learning,” Computers and Electronics in Agriculture, vol. 198, art. 107093, 2022, doi: 10.1016/j.compag.2022.107093.

[4] H. Wang et al., “A deep evidence fusion framework for apple leaf disease classification,” Engineering Applications of Artificial Intelligence, vol. 136, art. 109011, 2024, doi: 10.1016/j.engappai.2024.109011.

[5] H. Guan et al., “A lightweight model for efficient identification of plant diseases and pests based on deep learning,” Frontiers in Plant Science, vol. 14, art. 1227011, 2023, doi: 10.3389/fpls.2023.1227011.

[6] L. Gao et al., “A Lightweight YOLOv8 Model for Apple Leaf Disease Detection,” Applied Sciences, vol. 14, no. 15, art. 6710, 2024, doi: 10.3390/app14156710.

[7] M. Iftikhar, I. A. Kandhro, N. Kausar, A. Kehar, M. Uddin, and A. Dandoush, “Plant disease management: A fine-tuned enhanced CNN approach with mobile app integration for early detection and classification,” Artificial Intelligence Review, vol. 57, art. 167, 2024, doi: 10.1007/s10462-024-10809-z.

[8] A. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C.-W. Lin, “AFD-Net: Apple Foliar Disease multi-classification using deep learning on plant pathology dataset,” Plant and Soil, vol. 477, pp. 595–611, 2022, doi: 10.1007/s11104-022-05407-3.

[9] Y. Gao, Z. Cao, W. Cai, G. Gong, G. Zhou, and L. Li, “Apple Leaf Disease Identification in Complex Background Based on BAM-Net,” Agronomy, vol. 13, no. 5, art. 1240, 2023, doi: 10.3390/agronomy13051240.

[10] N. Parashar and P. Johri, “Enhancing Apple Leaf Disease Detection: A CNN-based Model Integrated with Image Segmentation Techniques for Precision Agriculture,” International Journal of Mathematical, Engineering and Management Sciences, vol. 9, no. 4, pp. 943–964, 2024, doi: 10.33889/IJMEMS.2024.9.4.050.

[11] H. Si, M. Li, W. Li, G. Zhang, M. Wang, F. Li, and Y. Li, “A Dual-Branch Model Integrating CNN and Swin Transformer for Efficient Apple Leaf Disease Classification,” Agriculture, vol. 14, no. 1, art. 142, 2024, doi: 10.3390/agriculture14010142.

[12] M. M. Alqahtani et al., “Sailfish Optimizer with EfficientNet Model for Apple Leaf Disease Detection,” Computers, Materials & Continua, vol. 74, no. 1, pp. 217–233, 2023, doi: 10.32604/cmc.2023.025280.

[13] A. Ait Nasser and M. A. Akhloufi, “A Hybrid Deep Learning Architecture for Apple Foliar Disease Detection,” Computers, vol. 13, no. 5, art. 116, 2024, doi: 10.3390/computers13050116.

[14] M. U. Ali et al., “AppleLeafNet: A lightweight and efficient deep learning framework for diagnosing apple leaf diseases,” Frontiers in Plant Science, vol. 15, art. 1502314, 2024, doi: 10.3389/fpls.2024.1502314.

[15] B. Liu et al., "MCDCNet: Multi-scale constrained deformable convolution network for apple leaf disease detection," Computers and Electronics in Agriculture, vol. 222, art. 109028, 2024, doi: 10.1016/j.compag.2024.109028.

[16] S. Zhu, W. Ma, J. Lu, B. Ren, C. Wang, and J. Wang, "A novel approach for apple leaf disease image segmentation in complex scenes based on two-stage DeepLabv3+ with adaptive loss," Computers and Electronics in Agriculture, vol. 204, art. 107539, 2023, doi: 10.1016/j.compag.2022.107539.

[17] H. Zou, P. Lv, and M. Zhao, "Detection of Apple Leaf Diseases Based on LightYOLO-AppleLeafDx," Plants, vol. 14, no. 4, art. 599, 2025, doi: 10.3390/plants14040599.

[18] C. Zhou et al., "A Multimodal Parallel Transformer Framework for Apple Disease Detection and Severity Classification with Lightweight Optimization," Agronomy, vol. 15, no. 5, art. 1246, 2025, doi: 10.3390/agronomy15051246.

[19] D. Adithiyaa, T. Lakshhmi Narayanan, and M. R. Prusty, "An attention-augmented lightweight convolutional framework for fine-grained plant leaf disease classification," Frontiers in Plant Science, vol. 17, art. 1762956, 2026, doi: 10.3389/fpls.2026.1762956.

[20] M. Thanjaivadivel, C. Gobinath, J. Vellingiri, S. Kaliraj, and J. S. Femilda Josephin, "EnConv: enhanced CNN for leaf disease classification," Journal of Plant Diseases and Protection, vol. 132, art. 32, 2025, doi: 10.1007/s41348-024-01033-6.

[21] M. F. Ahamed, A. Salam, M. Nahiduzzaman, M. Abdullah-Al-Wadud, and S. M. R. Islam, "Streamlining plant disease diagnosis with convolutional neural networks and edge devices," Neural Computing and Applications, vol. 36, pp. 18445–18477, 2024, doi: 10.1007/s00521-024-10152-y.

[22] K. Taji, A. Sohail, T. Shahzad, B. S. Khan, M. A. Khan, and K. Ouahada, "An Ensemble Hybrid Framework: A Comparative Analysis of Metaheuristic Algorithms for Ensemble Hybrid CNN Features for Plants Disease Classification," IEEE Access, vol. 12, pp. 61886–61906, 2024, doi: 10.1109/ACCESS.2024.3389648.

[23] A. Majid et al., "An Integrated Deep Learning Framework for Fruits Diseases Classification," Computers, Materials & Continua, vol. 71, no. 1, pp. 1387–1402, 2022, doi: 10.32604/cmc.2022.017701.

[24] S. Rehman et al., "Fruit Leaf Diseases Classification: A Hierarchical Deep Learning Framework," Computers, Materials & Continua, vol. 75, no. 1, pp. 1179–1194, 2023, doi: 10.32604/cmc.2023.035324.

[25] H. N. Ngugi, A. A. Akinyelu, and A. E. Ezugwu, "Machine Learning and Deep Learning for Crop Disease Diagnosis: Performance Analysis and Review," Agronomy, vol. 14, no. 12, art. 3001, 2024, doi: 10.3390/agronomy14123001.

[26] E. Kannan et al., "Deep Learning Techniques Advancements in Apple Leaf Disease Detection," Procedia Computer Science, vol. 235, pp. 713–722, 2024, doi: 10.1016/j.procs.2024.04.068.

[27] S. Karasu, "Apple Scab Classification Using 2D Shearlet Transform with Integrated Red Deer Optimization Technique in Convolutional Neural Network Models," Electronics, vol. 14, no. 23, art. 4678, 2025, doi: 10.3390/electronics14234678.

[28] S. Natarajan, P. Chakrabarti, and M. Margala, "Robust diagnosis and meta visualizations of plant diseases through deep neural architecture with explainable AI," Scientific Reports, vol. 14, art. 13695, 2024, doi: 10.1038/s41598-024-64601-8.

[29] S. M. Alhammad, D. S. Khafaga, W. M. El-hady, F. M. Samy, and K. M. Hosny, "Deep learning and explainable AI for classification of potato leaf diseases," Frontiers in Artificial Intelligence, vol. 7, art. 1449329, 2025, doi: 10.3389/frai.2024.1449329.

Confidence-Gated Indonesian Grammatical Error Correction Under Synthetic-to-Authentic Domain Shift

Eko Suroso[1]*, Onok Yayang Pamungkas[2], Laily Nurlina[3], Sudiyono[4], Jimat Susilo[5], Eko Priyanto[6], Hera Septriana[7]
Indonesian Language and Literature Education, Universitas Muhammadiyah Purwokerto, Banyumas, Indonesia[1, 2, 3, 7]
Social Sciences and Humanities Research Center, National Research and Innovation Agency of the Republic of Indonesia, Indonesia[4]
Indonesian Language Education, Universitas Swadaya Gunung Jati, Cirebon, Indonesia[5]
Pancasila and Citizenship Education, Universitas Muhammadiyah Purwokerto, Banyumas, Indonesia[6]

***Abstract*—Indonesian grammatical error correction (GEC) requires decisions about when to apply a proposed edit under synthetic-to-authentic domain shift. We evaluate a selective correction framework using a reconstructed, fully executable experiment. An audit of 1,955 authentic records removed 36 incomplete pairs, 14 duplicates, 45 string identities, and five additional token identities, leaving 1,855 pairs. Exact-text groups were partitioned into 1,297 training, 280 validation, and 278 test pairs. Synthetic priors were learned from 200,000 public development pairs. A cross-fitted logistic ranker combined synthetic mappings, authentic templates, spelling retrieval, and local language-model evidence. Thresholds were chosen on validation data under joint degradation and overcorrection budgets. At the 2% budget, the gate accepted five source edits, four correct: precision 80.00%, F0.5 4.03, coverage 1.80%, and word-error reduction 0.88%. No test source was degraded, but one of 278 correct-target controls changed. The precision interval was 37.55–96.38%. Removing the synthetic channel reduced available gold candidates from 110 to 109 and accepted true positives from four to three. Replacing the language-model feature with frozen IndoBERT produced 66.67% precision at 2.16% coverage, with one degraded source and one changed control. The results support restrictive authorization on this corpus, show limited unique synthetic contribution, and expose a substantial candidate-recall bottleneck. They do not establish broad correction ability or readiness for educational deployment.**

Keywords*—Indonesian grammatical error correction; domain shift; synthetic data; selective prediction; abstention; overcorrection; trustworthy natural language processing; writing feedback

I. Introduction

Grammatical error correction (GEC) is the computational task of detecting and repairing errors in written text, including orthographic, morphological, syntactic, lexical, and meaning-related problems [1]. The task is technically demanding because a system must not merely generate fluent text. It must identify which segment is erroneous, make the smallest defensible intervention, preserve the writer's intended meaning, and leave already acceptable text unchanged. These requirements are especially important in educational settings. A false correction may teach an incorrect rule, erase a legitimate stylistic choice, or encourage learners to surrender judgment to an automated system. For Indonesian-language education, reliable GEC could support formative feedback in academic writing, editing practice, and self-regulated revision. Yet an educationally safe system must recognize when it lacks sufficient evidence and defer to a teacher or human editor.

Indonesian GEC has progressed from a correction framework [2] to a Transformer supported by synthetic training pairs [3] and a framework for constructing evaluation corpora [4]. International work addresses complementary limitations: contextual augmentation targets error-distribution mismatch [5], RobustGEC tests sensitivity to contextual perturbations [6], and NaSGEC evaluates domain variation [7]. These developments motivate a separate question for writing assistance: when should a proposed correction actually be applied?

Synthetic corpora provide frequent spelling substitutions, character confusions, and surface edits, but their distribution reflects the corruption procedure. Authentic writing can require phrase restructuring, lexical interpretation, or several interacting changes. Wang et al. [5] combine contextual augmentation with relabeling-based cleaning; MixEdit [8] examines augmented-data affinity and diversity; and Stahlberg and Kumar [9] use tagged corruption to control synthetic error types. These approaches motivate testing distributional compatibility rather than assuming that more synthetic pairs guarantee reliable transfer.

Evaluation on error-containing sentences alone does not measure the cost of changing acceptable text. A selective writing assistant must therefore distinguish applying an edit from withholding it. In this implementation, correction is an accepted edit, retention describes unchanged output, and abstention describes withholding a proposed edit because its score is insufficient. Retention does not certify that a sentence is grammatical. The broader interface concept of correct, retain, or defer must not be interpreted as an evaluated three-class classifier. Varshney et al. [10] show that selective prediction performance can vary across tasks and distribution shifts, motivating direct evaluation of both erroneous inputs and correct-target controls.

This study combines synthetic replacement evidence from the Indonesian Transformer project [3] with the public GEC-GEE release [11]. It measures direct transfer, candidate-source ablations, budget sensitivity, score-tail calibration, and a frozen

*Corresponding author.

contextual scoring alternative. The paired evaluation processes every erroneous source and its corrected target as a no-error control. This makes unnecessary changes observable alongside successful repairs.

For this revision, we reconstructed the computational pipeline because the original executable artifacts and split indices could not be recovered from the submission files. The experiments below constitute a new evaluation and supersede the previous performance values. The audit, tokenizer, ontology rules, grouped partitions, model settings, candidate logs, and policy selections are now explicitly recorded. The historical 4.66% Identity exact-match value and the omitted category case cannot be traced to original predictions; both reporting problems are addressed through consistent token-level filtering and complete denominators in the reconstructed benchmark.

Four questions guide the revised evaluation: what marginal contribution do synthetic candidates make within the combined model (RQ1); how much observed harm is reduced by authorization at controlled coverage (RQ2); which error families and pipeline stages limit useful correction (RQ3); and how do alternative harm budgets and contextual scoring affect the operating point (RQ4)? The contribution is a reproducible evaluation protocol with paired controls and explicit abstention, rather than a claim of a new general-purpose GEC generator.

II. Related Work

A. Modern Grammatical Error Correction

The survey by Bryant et al. [1] organizes sequence generation, tagging, edit-based modeling, and evaluation. MainGEC [12] weights training examples using annotation properties; MixEdit [8] studies augmentation affinity and diversity; TransGEC [13] uses translated text to reduce the style mismatch in augmentation; and Mesham et al. [14] extend the sequence-tagging vocabulary for spelling and morphology. Cao et al. [15] introduce non-autoregressive signals into autoregressive correction. Luhtaru et al. [16] investigate multilingual transfer using pretrained translation models, while MoECE [17] uses expert specialization to improve efficiency and interpretability. These methods concern how candidate corrections are learned or generated; they do not establish the safety of the present authorization policy.

Fei et al. [18] incorporate explanations into GEC systems, while GEE! [19] evaluates whether large language models can locate and explain errors. Peng et al. [20] retrieve demonstrations using grammatical-error representations rather than semantic similarity alone. Li et al. [21] use explanations for demonstration retrieval. These approaches can strengthen candidate support, but an explanation must still be assessed separately from whether the proposed edit is correct.

B. Synthetic Data and Domain Mismatch

Wang et al. [5] combine rule-based substitution, context generation, and relabeling to improve synthetic examples. Stahlberg and Kumar [9] generate low-resource training data with tagged corruption models. In Bangla, Bhattacharyya and Bhattacharya [22] define error categories and use rule-based noise for instruction-tuning data. Smruti [23] combines a language model with non-parametric memory for Gujarati and supplies an evaluation dataset. Team Horizon [24] combines synthetic augmentation with multilingual Transformer models for Indic GEC. These are distinct data and modeling strategies; their reported benefits do not directly establish transfer to the Indonesian release evaluated here.

For Indonesian, Musyafa et al. [3] provide a synthetic corpus for Transformer-based correction. Lin et al. [4] propose a separate corpus-construction framework. The authentic data audited here are the Indonesian GEC-GEE release maintained in the syauqiezjut repository [11]; they should not be conflated with the corpus released with [4]. Frequent synthetic substitutions can generate plausible candidates, but their incremental contribution to correct accepted edits must be measured by removing the synthetic channel while retaining the remaining gated pipeline.

C. Robustness and Overcorrection

RobustGEC [6] examines sensitivity to minor contextual perturbations, and NaSGEC [7] documents cross-domain variation. Chang and Zhu [25] use chunk-based prompting to limit unnecessary correction. LM-Combiner [26] employs contextual rewriting to combine correction outputs. Liu et al. [27] introduce a chain-of-task instruction-tuning framework, while Wang et al. [28] investigate multi-pass decoding. These approaches motivate explicit control tests because improved correction or generation scores need not imply fewer unnecessary changes.

D. Evaluation and Selective Prediction

The gec-metrics library [29] provides a common interface for GEC evaluation. IMPARA-GED [30] incorporates grammatical-error detection into reference-free quality estimation, whereas Goto et al. [31] examine reliability problems in reference-free metrics. Varshney et al. [10] compare selective prediction across distribution settings; their subsequent method [32] calibrates predictions using confidence and instance difficulty. Zablotskaia et al. [33] evaluate uncertainty and selective generation for summarization, and Ba et al. [34] use synthetic data for calibration. Guo et al. [35] study post-hoc calibration of neural networks. These findings support reporting ranking quality, calibration, coverage, and observed harm as separate quantities.

E. Indonesian NLP and Educational Writing Support

IndoNLU [36] supplies language-understanding benchmarks and models, and IndoNLG [37] supplies benchmarks for language generation. IndoBERTweet [38] adapts a pretrained encoder to Indonesian Twitter, IndoNLI [39] provides natural-language inference data, and Cendol [40] develops instruction-tuned generative models for Indonesian languages. KEBI [41] addresses spelling detection in scientific papers, while U-Tapis [42] evaluates spelling support for journalism students. Mohsen [43] synthesizes computer-mediated corrective-feedback studies; Zhai and Ma [44] synthesize automated writing evaluation; Fan [45] examines automated feedback in an EFL writing study; and Lee and Moore [46] review generative-AI feedback in higher education. These studies motivate supervised feedback research but do not demonstrate learning benefits for the present system. Koto et al. [47] introduce IndoLEM and a pretrained Indonesian BERT model; the present comparison uses its frozen masked-language-model checkpoint.

III. Data Sources and Quality Audit

A. Synthetic Source

Musyafa et al. [3] report a large synthetic corpus generated from Indonesian text. The accessible development and test CSV archives contained 424,430 and 424,419 parsable pairs, respectively, totaling 848,849. Their 47 string identities and 67 token identities differ because tokenization ignores spaces adjacent to punctuation. The archives are audited separately; the synthetic test partition contributes no training evidence.

Using Python random seed 2026, we sampled 200,000 development rows without replacement after checking exact normalized-text overlap with the authentic corpus. No overlapping development pair was found. Minimum-edit alignment supplied single-token replacement counts. A replacement was retained at support ≥2 and conditional probability ≥0.50 among nonidentity substitutions of its source token, with at most three alternatives per source. This yielded 40,147 retained replacements for 39,985 source tokens. Sample row identifiers and hashes of all three source CSVs are included in the computational supplement.

B. Authentic Source

The GEC-GEE repository [11] describes 1,920 expert-validated pairs from academic publications dated 2020–2024, whereas its downloadable CSV contains 1,955 records. It supplies source, target, label, and explanation fields. It does not supply row-level genre, writer proficiency, register, or document provenance. We therefore describe its stated origin without assigning unsupported demographic or genre proportions. Its scope is one heterogeneous public release.

We normalized text with Unicode NFKC, collapsed whitespace, removed blank source/target pairs, retained the first duplicate normalized pair, and removed remaining string identities. These sequential exclusions were 36, 14, and 45, yielding 1,860 string-distinct pairs. The case-preserving tokenizer matches Unicode words and individual nonspace punctuation with the regular expression \w+|[^\w\s]. Five additional pairs were identical under these tokens and were explicitly removed, leaving 1,855 pairs. No additional token-pair duplicate remained. Training, exact match, edit extraction, and distance evaluation all use these same token sequences, so Identity has zero exact match on the error benchmark.

C. Ontology and Grouped Partitioning

Existing labels were mapped by published, case-normalized keyword rules to orthography, punctuation, morphology, function word, redundancy, lexical choice, semantic coherence, and syntax, with Other for residual labels. Typographical variants were explicitly standardized. Primary-family priority follows that order; multiple labels are retained for descriptive purposes. For example, afiksasi and reduplikasi indicate morphology, preposisi and konjungsi indicate function words, and pleonasme indicates redundancy. These are operational groupings of existing annotations, not newly validated linguistic judgments.

Exact token-identical source or target texts were joined into 1,845 connected groups. A seeded search over 400 group partitions minimized size and category-count imbalance relative to a 70/15/15 split, using no model outcomes. The selected partitions contain 1,297 training, 280 validation, and 278 test pairs, with no exact text shared between partitions. Their row IDs and complete label-to-family mappings are supplied. Across the final corpus, sources average 14.00 tokens, with median 12 and 95th percentile 26.3; minimum-edit alignment yields 3,159 gold spans. The new counts supersede the historical tokenizer and split statistics.

TABLE I. Sequential Cleaning with Consistent Token Evaluation

Sequential stage	Entering	Removed	Remaining
Release	1,955	0	1,955
Remove empty source or target	1,955	36	1,919
Deduplicate normalized pairs	1,919	14	1,905
Remove string identities	1,905	45	1,860
Remove additional token identities	1,860	5	1,855
Deduplicate token pairs	1,855	0	1,855

Table I accounts for all 100 exclusions without adding overlapping audit categories. The distinction between string identity and token identity is explicit. This prevents an unchanged source from being counted as an exact correction merely because evaluation applies a more permissive normalization. Dataset quality and preprocessing decisions affect downstream inference and must be auditable [48], [49], [50].

IV. Proposed Framework

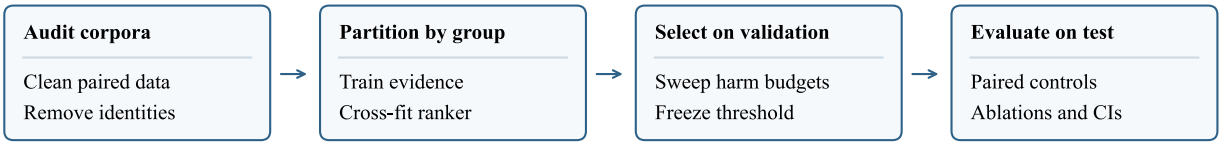
A. Design Principle

Candidate discovery proposes a local change; ranking orders proposals; authorization determines which changes are applied. Correction means an accepted edit. Retention describes unchanged output, while abstention records rejection below the score threshold. Overlap and the five-edit cap are separate rejection reasons; no-candidate retention is also logged. An unchanged output does not certify grammatical correctness. The implementation is therefore an edit-selection policy with rejection records, not an evaluated three-class correct/retain/abstain classifier. Fig. 1 shows the full experimental flow.

B. Tokenization and Alignment

Case-preserving tokens are aligned by unit-cost Levenshtein operations [51] using RapidFuzz 3.14.6. Consecutive non-equal operations are merged into a single span; exact matches use source start/end offsets and the replacement token tuple. Authentic templates are retained when each side contains at most three tokens, including insertion and deletion. Support is the number of training occurrences. Conditional probability divides

template support by all nonidentity rule support for the same source span; insertion rules share an empty-source denominator. This definition is available in executable code rather than inferred from reported metrics.



Candidate discovery → ranking → edit authorization

Fig. 1. Reconstructed data flow and separation of candidate discovery, ranking, and authorization.

C. Candidate Sources

Synthetic candidates follow the fixed support/probability filters in Section III. Authentic templates store distributions of the immediately adjacent left and right tokens, including sentence boundaries. Context agreement is half the sum of the two matching-neighbor proportions. An insertion requires at least one matching neighbor. Spelling retrieval searches only the current training-target vocabulary: alphabetic tokens of length ≥3 and frequency ≥2, Levenshtein distance ≤1 for source length ≤5 and ≤2 otherwise, normalized similarity ≥0.70, and at most three alternatives ordered by distance, descending frequency, then lexical order. No dictionary or reference target is consulted during inference.

D. Candidate Features and Language Model

Sixteen features encode log support; conditional probability; log source and target frequencies; their smoothed log ratio; character similarity; source and replacement token lengths; character-length difference; punctuation and case-change flags; three candidate-origin flags; context agreement; and local language-model gain. Duplicate candidates with the same offsets and replacement are merged while retaining all origin flags. Source and target frequencies use only the current authentic training fold. The language model uses lowercased target-training bigrams, BOS/EOS and UNK, with add-0.1 smoothing. Gain is the difference in mean log-likelihood within two neighboring tokens on either side. The complete feature order and formulas accompany the code.

E. Cross-Fitted Ranker

Five group-disjoint folds within the training partition induced rules and vocabulary on four folds and generated candidates for held-out sources and their target controls.

Candidate labels indicate exact gold-edit matches. This produced 207,580 candidates and 472 positives. StandardScaler and L2 logistic regression were fitted to these examples with C=1, balanced class weights, lbfgs, max_iter=2000, and seed 2026. The final evidence dictionaries used only the complete authentic training partition. Calibration metrics are reported on validation and test candidates, not on the same candidates used to fit the ranker.

F. Validation-Constrained Authorization

Thresholds cover 0.00–1.00 at increments of 0.01, plus 0.995, 0.999, and a retain-all option. Validation F0.5 is maximized subject to the chosen budget; ties favor lower degradation, lower overcorrection, and then a higher threshold. Budgets of 1%, 2%, and 5% are evaluated with overcorrection alone and with a joint bound on degradation and overcorrection. The primary condition is the full model at the joint 2% budget, fixed before inspecting its test outputs. This is an empirical tolerance for comparing policies, not a validated educational safety standard. A budget does not guarantee the same population or test risk.

G. Comparators and Ablations

Table II defines Identity, direct synthetic and authentic rules, a fixed heuristic, an ungated ranker, the primary gate, four ablations, and an IndoBERT feature replacement. Channel ablations remove both candidates unique to the channel and that channel's support/probability/origin evidence in merged candidates; the logistic ranker is refitted. Authentic language statistics remain training-only in every channel ablation. The no-LM condition sets the local gain feature to zero. All active policies apply at most five nonoverlapping edits in one pass, without iterative rewriting.

TABLE II. EVALUATED CORRECTION VARIANTS

Variant	Candidate evidence	Authorization policy
Identity	None	Retain every input
Synthetic direct	Synthetic channel	Probability/support ranking
Authentic direct	Authentic channel	Probability/support ranking
Heuristic	All channels	Fixed weighted score ≥0.50
Ungated ranker	All channels	Logistic score with no gate
Confidence gated	All channels	Validation-selected joint 2% gate
Channel ablations	Remove synthetic, authentic, or spelling	Refit and reselect independently
No n-gram feature	Same full candidate pool	Zero LM gain; refit and reselect
IndoBERT feature	Same full candidate pool	Replace LM gain; refit and reselect

Algorithm 1 summarizes the reconstructed inference procedure. The implementation code and per-sample numerical results are provided in the supplementary material described at the end of this paper.

Algorithm 1. Confidence-Gated Selective Correction

Input: source tokens x; training evidence E; fitted ranker f; validation-selected threshold τ

1. Generate local candidates from the permitted channels and deduplicate identical edits.
2. Compute inference features from x and training evidence only; preserve origin flags.
3. Score and sort candidates once, breaking ties deterministically by offsets and replacement.
4. Accept an edit only if its score ≥ τ, it does not overlap an accepted edit, and fewer than five edits have been accepted.
5. Log rejection as below threshold, overlap, or edit cap; distinguish no-candidate retention.
6. Apply accepted edits from right to left to preserve source offsets.
7. Return the output and rejection record. Retention does not certify correctness.

V. Experimental Design

A. Evaluation Units

The held-out test set contains 278 erroneous sources with 476 gold edit spans and 278 corresponding reference-correct controls. Validation has 280 paired sources/controls. Exact-text grouping prevents the same source or target from appearing in both training and evaluation. Shared words and rules remain possible, as expected for local correction; near-duplicate and provenance overlap are not ruled out. Each variant's threshold is selected on validation before its test evaluation.

B. Edit Metrics

A system edit is a true positive only if its source offsets and replacement tokens exactly equal a gold span. Precision equals TP/predicted edits; recall equals TP/gold edits; F0.5 = 1.25TP/(predicted edits + 0.25 gold edits). Scores are displayed on a 0–100 scale. A partly helpful change can lower token distance without matching a complete reference edit. When a system makes no edit, precision is mathematically undefined; tables mark it with a dash, while the evaluation convention sets F0.5 and recall to zero.

C. Sentence and Distance Metrics

Exact match compares output and target tokens. Coverage is the proportion of source sentences changed. Improvement and degradation compare output-to-target Levenshtein distance with the initial source-to-target distance. Overcorrection is any token change on a target control. Word-error reduction is one minus the ratio of aggregate output and initial token distances; character reduction uses the analogous distance on token-normalized strings. We distinguish unconditional degradation over all sources from conditional degradation among changed sources, and report both denominators.

D. Uncertainty and Comparisons

We report Wilson 95% intervals for rates [52], two-sided exact McNemar degradation comparisons [53], Holm adjustment across ten comparisons with the primary gate, and 10,000 paired sentence bootstrap samples with seed 2026 for token-distance gain [54]. These are descriptive intervals for this corpus; candidates from the same sentence can be dependent, so candidate-tail Wilson intervals do not establish independent-event calibration guarantees. A matched-coverage diagnostic restricts each ranked comparator to five changed source sentences and one edit per sentence. Two hundred seeded random authorizations provide an additional descriptive comparison, without influencing model selection.

E. Category Accounting and Bottlenecks

Every test pair appears once under its primary family, including Other. The analysis distinguishes gold edits that never enter the candidate pool from available gold edits rejected during authorization. Removing a candidate channel can change both the pool and its scoring evidence, so an end-to-end ablation effect is not automatically a uniquely generated edit. Accepted-candidate origin flags support a narrower attribution analysis.

F. Contextual Feature Comparison

We replace only the local n-gram gain with a frozen IndoBERT masked-span score [47], using indolem/indobert-base-uncased at revision 7ccb3cd0f5b08ffbaa465aade22328e8600e23eb. All edited subword tokens are masked simultaneously, and the feature is the change in their mean log probability between original and proposed spans. Windows use up to six neighboring word tokens per side, capped at 32 subwords per side and 16 edited subwords. Insertions/deletions include a shared neighboring token as an anchor. The trained MLM head is loaded with no missing weights. Candidates, other features, folds, and logistic settings are identical; no neural fine-tuning or candidate enrichment is performed.

G. Reproducible Implementation

The executed environment uses Python 3.12, NumPy 2.5.3, SciPy 1.18.1, scikit-learn 1.9.0, RapidFuzz 3.14.6, PyTorch 2.14.0, and Transformers 4.57.6. IndoBERT runs on CPU with four threads and batches of 32. The heuristic is fixed at 0.30 conditional probability + 0.25 context agreement + 0.20 similarity + 0.15 sigmoid(LM gain) + 0.10 support/(support+2), accepting scores ≥0.50. Direct-rule ties use support after conditional probability. The supplement records exact software versions, hashes, all split IDs, feature definitions, candidate counts, and validation sweeps.

H. Reconstruction and Leakage Controls

The revised protocol was written before model fitting and test-outcome inspection. It replaces undocumented historical choices with explicit settings and uses new grouped partitions. The full n-gram model is the prespecified primary condition, rather than the variant with the highest test score. No test label, explanation, or target enters rule induction, frequency estimation, logistic fitting, or threshold selection. Exact-text overlap checks use corpus strings only for duplicate detection. Unrecoverable historical calibration variants and their test-selected comparison have been removed, and the new values are disclosed as a reconstruction rather than exact reproduction.

VI. Results

A. Main Comparison

Table III presents accuracy and intervention outcomes separately. Identity correctly has zero exact matches. Direct synthetic application changes 63.67% of sources, degrades 53.60%, and changes 58.99% of controls. Its word-error reduction is -25.83%, demonstrating harmful direct transfer in this implementation. These values supersede those obtained with the unrecovered historical preprocessing and candidate filters.

Authentic direct rules change every source and every control, with 95.32% degradation. The ungated ranker degrades 93.17% and also changes every control. The heuristic has the higher unconstrained F0.5 of 8.03, but degrades 53.24% and overcorrects 62.23%. A higher conventional edit score therefore does not imply an acceptable intervention policy.

The primary gate selects threshold 0.999 and changes five sources, with four of five system edits matching the reference. Precision is 80.00%, recall 0.84%, and F0.5 4.03. Coverage is 1.80%, word-error reduction 0.88%, and one source is corrected exactly. All five changed sources become closer to their targets; none degrades. One correct-target control changes (0.36%), so the revised results do not support the earlier claim of zero observed overcorrection.

TABLE III. Reconstructed Test Performance on 278 Sources and 278 Controls (Panel A Edit and Sentence Accuracy)

System	Precision (%)	Recall (%)	F0.5	Exact match (%)
Identity	—	0.00	0.00	0.00
Synthetic direct	2.01	1.26	1.79	0.72
Authentic direct	2.31	5.04	2.59	0.00
Heuristic	8.83	5.88	8.03	3.24
Ungated ranker	2.99	6.30	3.34	0.00
Confidence gated	80.00	0.84	4.03	0.36
No synthetic channel	75.00	0.63	3.05	0.36
No authentic channel	18.18	0.84	3.55	0.36
No spelling channel	—	0.00	0.00	0.00
No n-gram feature	—	0.00	0.00	0.00
IndoBERT feature	66.67	0.84	4.00	0.00

Panel B Sentence Intervention and Harm

System	Degradation (%)	Coverage (%)	Word error reduction (%)	Overcorrection (%)
Identity	0.00	0.00	0.00	0.00
Synthetic direct	53.60	63.67	-25.83	58.99
Authentic direct	95.32	100.00	-114.90	100.00
Heuristic	53.24	70.86	-23.51	62.23
Ungated ranker	93.17	100.00	-102.87	100.00
Confidence gated	0.00	1.80	0.88	0.36
No synthetic channel	0.00	1.44	0.77	0.36
No authentic channel	5.04	7.55	-1.10	5.04
No spelling channel	0.00	0.00	0.00	0.00
No n-gram feature	0.00	0.00	0.00	0.00
IndoBERT feature	0.36	2.16	0.66	0.36

B. Safety and Useful Intervention

Table IV reports the uncertainty in the low-intervention results. The primary precision interval is 37.55–96.38%. Unconditional degradation is 0/278, with an upper bound of 1.36%; conditional degradation is 0/5 changed sources, whose Wilson upper bound is 43.45%. Overcorrection is 1/278, with interval 0.06–2.01%. These denominators prevent the unconditional zero-degradation observation from being interpreted as evidence of near-perfect reliability for accepted corrections.

The primary mean token-distance gain is 0.0288 tokens per source, with bootstrap interval 0.0072–0.0576. Paired harm tests favor the gate over the aggressive direct and ungated systems, but those comparisons alone confound authorization with coverage. At a fixed five changed sources and one edit each, the full ranking retains four correct edits without degradation. Random selection averages 0.60% precision and 1.61% unconditional degradation over 200 runs. The matched diagnostic is small and descriptive; it does not establish a general safety guarantee or universal superiority at all coverage levels.

Fig. 2 separates coverage from the two harm rates. The primary gate improves observed safety through very restrictive selection. Identity attains no intervention-related harm by doing nothing, so the gain in useful corrections and its uncertainty must accompany any safety comparison.

TABLE IV. UNCERTAINTY AND PAIRED DEGRADATION COMPARISONS

System	Source harm	Controls changed	Precision 95% CI	Mean gain 95% CI	Adjusted p
Synthetic direct	149/278	164/278	0.92–4.31%	-0.9676–-0.7194	2.2e-44
Authentic direct	265/278	278/278	1.56–3.41%	-3.9496–-3.5396	3.4e-79
Heuristic	148/278	173/278	6.18–12.47%	-0.9245–-0.6151	3.9e-44
Ungated ranker	259/278	278/278	2.10–4.23%	-3.5755–-3.1259	1.9e-77
Confidence gated	0/278	1/278	37.55–96.38%	0.0072–0.0576	—
IndoBERT feature	1/278	1/278	30.00–90.32%	-0.0072–0.0504	1

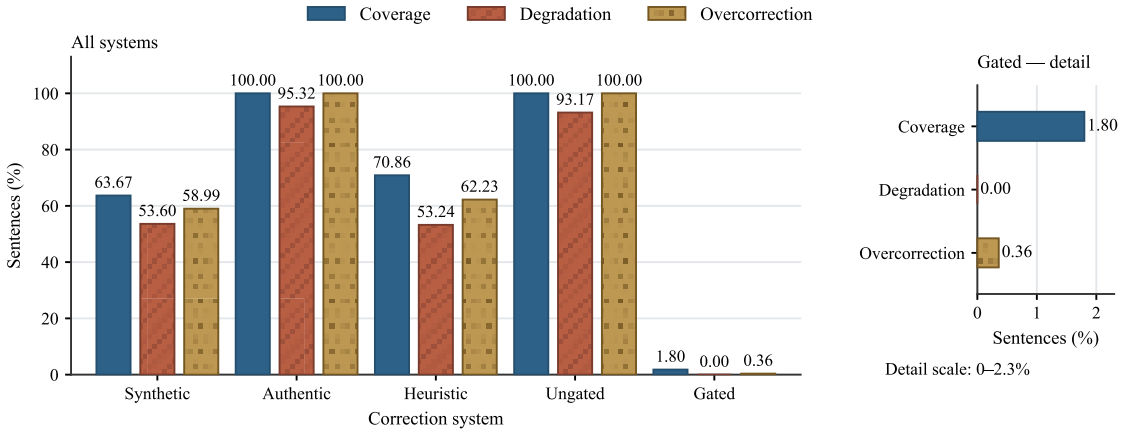


Fig. 2. Sentence-level harm and benefit under synthetic-to-authentic transfer.

C. Budget Sensitivity

Under the full n-gram model, both the overcorrection-only and joint formulations select retain-all at 1%, threshold 0.999 at 2%, and 0.995 at 5%. The resulting test coverage is 0%, 1.80%, and 7.91%; F0.5 is 0, 4.03, and 8.87. At 5%, test degradation rises to 3.60% and overcorrection to 4.68%. Fig. 3 plots validation frontiers; the full validation/test sweep and six selected policies per model are supplied. The 280 validation controls permit at most 2, 5, or 14 changed controls under the three budgets. These are empirical tolerances, not confidence-bound guarantees.

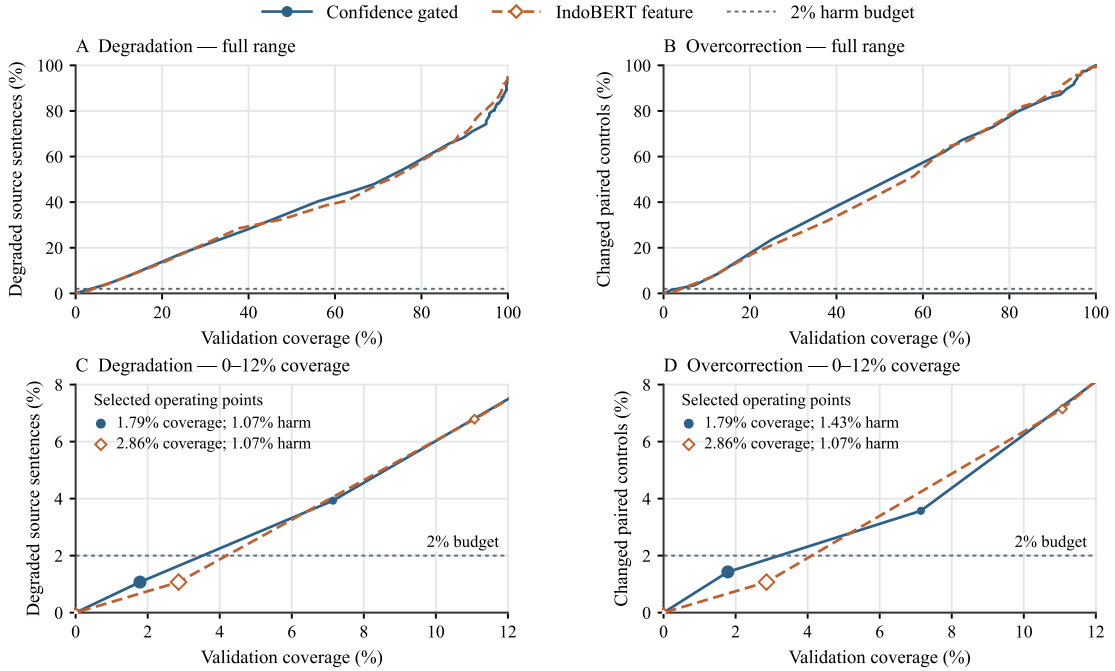


Fig. 3. Validation harm versus coverage for n-gram and IndoBERT scoring. Dots mark the joint 2% policies.

D. Error Families

Table V and Fig. 4 exhaust all 278 test sources: orthography 78, punctuation 15, morphology 56, function word 19, redundancy 23, lexical choice 42, semantic coherence 15, syntax 27, and Other 3. Correct edits occur in orthography (two), redundancy (one), and syntax (one). A lexical-group edit improves distance without matching a complete gold span. The control overcorrection occurs in the morphology group. These family labels describe each pair's primary annotation, not necessarily the linguistic type of its accepted edit. The earlier inference that all structural categories were deferred is therefore withdrawn.

TABLE V. COMPLETE PRIMARY-FAMILY ACCOUNTING FOR THE GATED SYSTEM

Family	n	TP/edits	Recall (%)	F0.5	Coverage (%)	Controls changed (%)
Orthography	78	2/2	1.38	6.54	2.56	0.00
Punctuation	15	0/0	0.00	0.00	0.00	0.00
Morphology	56	0/0	0.00	0.00	0.00	1.79
Function word	19	0/0	0.00	0.00	0.00	0.00
Redundancy	23	1/1	2.63	11.90	4.35	0.00
Lexical	42	0/1	0.00	0.00	2.38	0.00
Semantic	15	0/0	0.00	0.00	0.00	0.00
Syntax	27	1/1	1.92	8.93	3.70	0.00
Other	3	0/0	0.00	0.00	0.00	0.00

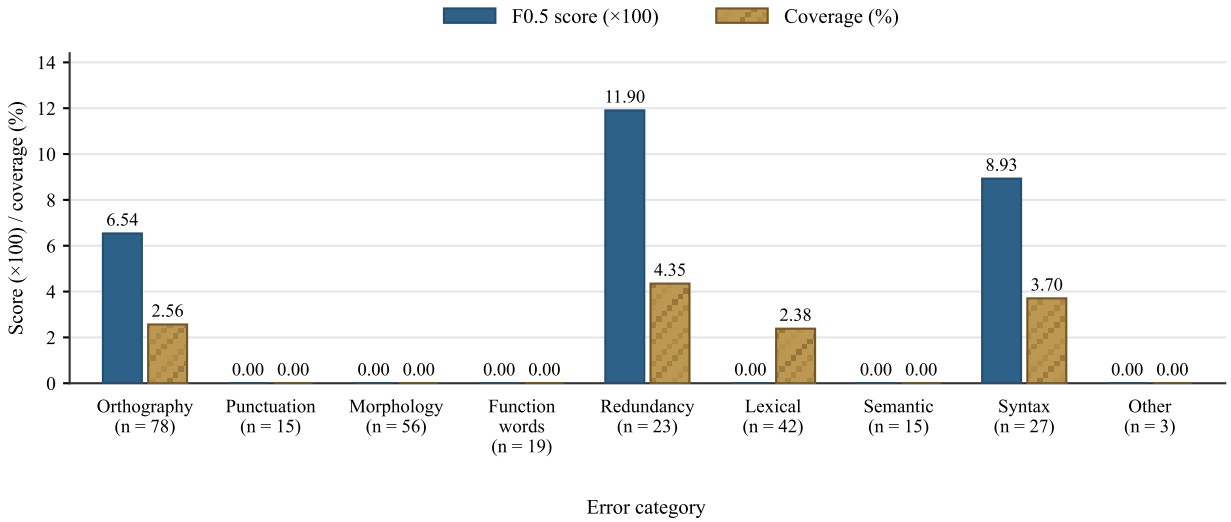


Fig. 4. Category-specific F0.5 and coverage for the confidence-gated system.

E. Candidate Attribution

The full pool contains 110 of 476 gold edits (23.11%), across 84 of 278 source sentences. Without synthetic candidates, 109 gold edits remain available, indicating only one uniquely supplied gold candidate. Of the four accepted gold candidates, three originate only from authentic templates and one is shared by authentic and synthetic channels. Removing synthetic evidence and refitting yields three true positives from four edits, F0.5 3.05 and coverage 1.44%. The one-edit end-to-end change may reflect scoring support; it is not proof that an accepted correction was unavailable without synthetic generation. The revised RQ1 conclusion is therefore limited marginal contribution on this split.

F. Additional Ablations

Removing the authentic channel reduces the gold-candidate ceiling to 26/476 (5.46%). Its validation-selected policy changes 21 test sources and degrades 14, illustrating that an empirical validation budget may fail to transfer even within the same release. Removing spelling retrieval leaves 96/476 gold candidates, but its joint 2% selection is retain-all. The no-LM condition also selects retain-all. These outcomes describe the fixed candidate filters, feature representation, grid, and split; they do not establish that the removed components are universally necessary.

VII. DISCUSSION

A. Synthetic Scale and Generality

The synthetic source provides frequent surface substitutions, but only one unique exact gold candidate beyond the remaining channels on this test set. Its support can still influence a shared candidate's ranking. The controlled ablation therefore narrows the earlier claim that scale benefits candidate discovery: large synthetic resources are insufficient evidence of useful marginal

transfer. Authentic templates supply most of the exact gold-candidate coverage in this reconstructed system.

B. Ranking and Tail Calibration

The n-gram ranker has validation AUC 0.925, average precision 0.047, Brier score 0.100, and ECE 0.181. On test candidates, these are 0.937, 0.093, 0.101, and 0.184. Among the 205 source/control candidates scoring ≥0.98, only 20 match gold edits (9.76%), despite a mean score of 0.989. At ≥0.999, four of six candidates are correct, with interval 30.00–90.32%; one of the two errors is a control edit. The gate succeeds by severe empirical filtering rather than demonstrated probability calibration. A score of 0.999 must not be read as a 99.9% chance of correctness.

C. Coverage and Conditional Risk

Only 110 gold edits enter the full candidate pool, and only four are recovered by the selected policy. Thus, 366/476 gold edits are unavailable to authorization at any threshold. All test sources have at least one candidate, so unchanged output is caused by rejection rather than evidence that a sentence is correct. The small number of accepted edits gives weak precision and conditional-harm estimates. Matched-coverage diagnostics reduce one confound but remain exploratory and depend on the supplied candidate universe.

D. Educational Implications

Mohsen [43] and Zhai and Ma [44] synthesize evidence about corrective feedback and automated writing evaluation, respectively; Fan [45] studies revision with automated feedback, and Lee and Moore [46] review generative feedback in higher education. The current study measures computational outcomes only. A proposed teacher-mediated interface would display candidate evidence and distinguish accepted edits from uncertain retained spans. It should not label every unchanged sentence as correct. Teacher workload, learner understanding, and writing improvement require prospective evaluation.

E. Contextual Scoring

Replacing the n-gram feature with frozen IndoBERT [47] keeps the same candidate pool and fits the same logistic architecture. At the joint 2% policy, its threshold is 0.999, with four correct edits from six, precision 66.67%, F0.5 4.00, and coverage 2.16%. It degrades one source and changes one control, each by 0.36%. Its test ECE is 0.192. The experiment provides no material safety-coverage improvement in this configuration. It evaluates local frozen masked-span scoring, not full-sentence neural generation or a fine-tuned contextual corrector; the small event counts cannot establish equivalence between the alternatives.

F. Measured Technical Bottlenecks

Candidate recall is bounded at 23.11% before authorization, reflecting short-span templates, sparse authentic evidence, and a limited synthetic match to the corpus. Gating then recovers only 4/110 available gold edits. Improving only the ranking feature cannot recover gold edits absent from the pool. Future experiments should therefore separate richer generation from better authorization, maintain correct-target controls, and report both candidate recall and policy-level harm. Neural generator enrichment was not tested here.

G. Criteria for a Supervised Pilot

The present system remains below a defensible deployment threshold. A concrete future screening rule is at least 10% coverage on an independently sampled corpus, at least 90% edit precision by the lower 95% interval, and upper 95% bounds below 1% for both degradation and control overcorrection. These are proposed engineering targets for discussion with teachers, not standards or thresholds validated by this experiment. The current 1.80% coverage and broad precision interval fail that screen. A pilot must also measure correct accepted feedback per teacher-review minute, teacher acceptance, and learner understanding against the existing review process; coverage alone is insufficient.

H. Comparison at Equal Intervention

With exactly five changed source sentences and one edit each, the full ranking yields four correct edits and no degradation. The heuristic also yields four correct edits, but degrades one source; synthetic ranking yields one correct edit and three degraded sources, and authentic ranking yields no correct edit and five degraded sources. The full method therefore does not improve precision over the heuristic in this small matched comparison. These descriptive outcomes use frozen scores and do not replace the validation-selected policies in Table III. Random-authorized controls are not assessed in this source-only diagnostic, and the five-intervention sample cannot establish robust safety superiority.

VIII. Limitations and Threats to Validity

The first limitation is reconstruction fidelity: new explicit filters, tokenization, ontology mapping, grouped partitions, and ranking settings supersede unrecovered historical choices. New results cannot identify the exact cause of every historical number. Second, all operating-point results come from a single split with a fixed seed. Sentence bootstrap intervals reflect test-sampling variation conditional on that fitted system, not training-seed or corpus-to-corpus variability. Third, the primary gate makes five source edits, yielding broad precision and conditional-risk intervals. Low observed harm therefore cannot establish reliable operation at greater coverage.

Fourth, the GEC-GEE CSV [11] lacks row-level genre, register, proficiency, and document provenance. Its repository description does not establish representative sampling, and ontology mapping was not independently reannotated. Exact-text grouping limits one form of leakage but does not remove near-duplicates, shared source documents, or possible pre-trained-model exposure. Fifth, the synthetic corruption process represents one particular resource. Sixth, the IndoBERT comparison is a frozen local-feature replacement; broader contextual models, fine-tuning, and candidate enrichment may behave differently. Calibration must be re-evaluated under every changed distribution.

Seventh, target controls are reference-correct within the dataset, not universally error-free writing. A single reference may penalize legitimate alternative corrections. Independent no-error texts and multiple references would strengthen ecological validity. Eighth, the empirical 1%, 2%, and 5% budgets do not control unknown population risk, and the proposed pilot targets remain unvalidated. Finally, no teacher-workload, learning, or

longitudinal writing outcome is measured. Those claims require prospective educational evaluation after stronger computational evidence is established.

IX. Ethics, Data Availability, and AI Disclosure

This study analyzes publicly released text corpora and includes no new participant recruitment or intervention. The computational supplement contains executable code, source URLs and hashes, row identifiers, ontology mappings, aggregate statistics, and numeric per-sample outcomes. It does not redistribute third-party raw corpora or pretrained weights. Users should obtain those resources from their original providers and follow the corresponding terms. The source references identify the corpus and model used in the reconstruction.

Generative AI assisted with code development, methodological checking, drafting, and language revision. The reconstructed performance values are computed by executable scripts; unverified historical values are not retained as new experimental evidence. The authors remain responsible for checking the revised methods, data provenance, numerical results, and interpretations before submission. No gold correction or benchmark label was generated to improve the reported model performance.

Data and code availability. The accompanying reconstruction package includes preprocessing and model scripts, frozen split IDs and synthetic sample IDs, source hashes, validation selections, ablations, calibration tails, paired uncertainty analyses, figure generation, and verification checks. The package distinguishes the reconstructed evaluation from unrecovered historical artifacts. No funding or competing interests are declared.

X. Conclusion

The reconstructed experiment shows that aggressive local correction can introduce substantial harm on this public Indonesian corpus. Validation-constrained authorization reduces intervention to five of 278 sources, obtaining four exact gold edits, 80.00% precision, F0.5 4.03, and 0.88% word-error reduction. No source degrades, but one reference-correct control changes. All results use consistent token filtering and complete category denominators, and supersede the earlier unreproduced performance values.

The synthetic-channel ablation shows little uniquely added gold-candidate recall on this split. Budget sensitivity reveals a direct trade-off between useful coverage and observed harm, while the frozen IndoBERT replacement does not materially improve the selected operating point. The strongest remaining bottleneck is that most gold edits never enter the candidate pool. Selective authorization is therefore a useful research component, but improved generation, external evaluation, stronger uncertainty evidence, and teacher studies are required before the system can support claims of dependable educational deployment.

XI. Supplementary Material

Supplementary material for this article is available at https://drive.google.com/file/d/11qjUt0yG6TTwe0Nf_QEY5OybPaUcn3aX/view?usp=sharing. The package includes source code for data preprocessing, feature generation, policy selection, and evaluation, together with instructions for reproducing the experiments and the corresponding numerical results. Third-party datasets and pretrained model weights are not included. The accompanying documentation provides instructions for obtaining these resources from their original providers.

References

[1] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical Error Correction: A Survey of the State of the Art," Computational Linguistics, vol. 49, no. 3, pp. 643-701, 2023, doi: 10.1162/coli_a_00478.

[2] N. Lin, Y. Gao, C. Wu, and A. Musyafa, "A Framework for Indonesian Grammar Error Correction," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 4, art. 58, 2021, doi: 10.1145/3440993.

[3] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, "Automatic Correction of Indonesian Grammatical Errors Based on Transformer," Applied Sciences, vol. 12, no. 20, art. 10380, 2022, doi: 10.3390/app122010380.

[4] N. Lin, M. Zeng, W. Huang, S. Jiang, L. Xiao, and A. Yang, "A Simple Yet Effective Corpus Construction Framework for Indonesian Grammatical Error Correction," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 24, no. 1, art. 5, 2025, doi: 10.1145/3704264.

[5] Y. Wang, B. Wang, Y. Liu, Q. Zhu, D. Wu, and W. Che, "Improving Grammatical Error Correction via Contextual Data Augmentation," in Findings of the Association for Computational Linguistics: ACL 2024, pp. 10898-10910, 2024, doi: 10.18653/v1/2024.findings-acl.647.

[6] Y. Zhang, L. Cui, E. Zhao, W. Bi, and S. Shi, "RobustGEC: Robust Grammatical Error Correction Against Subtle Context Perturbation," in Proc. EMNLP 2023, pp. 16780-16793, 2023, doi: 10.18653/v1/2023.emnlp-main.1043.

[7] Y. Zhang, B. Zhang, H. Jiang, Z. Li, C. Li, F. Huang, and M. Zhang, "NaSGEC: A Multi-Domain Chinese Grammatical Error Correction Dataset from Native Speaker Texts," in Findings of ACL 2023, pp. 9935-9951, 2023, doi: 10.18653/v1/2023.findings-acl.630.

[8] J. Ye, Y. Li, Y. Li, and H.-T. Zheng, "MixEdit: Revisiting Data Augmentation and Beyond for Grammatical Error Correction," in Findings of EMNLP 2023, pp. 10161-10175, 2023, doi: 10.18653/v1/2023.findings-emnlp.681.

[9] F. Stahlberg and S. Kumar, "Synthetic Data Generation for Low-resource Grammatical Error Correction with Tagged Corruption Models," in Proc. BEA 2024, pp. 11-16, 2024.

[10] N. Varshney, S. Mishra, and C. Baral, "Investigating Selective Prediction Approaches Across Several Tasks in IID, OOD, and Adversarial Settings," in Findings of ACL 2022, pp. 1995-2002, 2022, doi: 10.18653/v1/2022.findings-acl.158.

[11] Syauqiezjut, "Indonesian Grammar Error Correction and Explanation Dataset," GitHub repository. Accessed: Sep. 10, 2026. [Online]. Available: https://github.com/syauqiezjut/indonesian-gec-gee

[12] J. Li, Q. Wang, C. Zhu, Z. Mao, and Y. Zhang, "Grammatical Error Correction via Mixed-Grained Weighted Training," in Findings of EMNLP 2023, pp. 6027-6037, 2023, doi: 10.18653/v1/2023.findings-emnlp.400.

[13] T. Fang, X. Liu, D. F. Wong, R. Zhan, L. Ding, L. S. Chao, D. Tao, and M. Zhang, "TransGEC: Improving Grammatical Error Correction with Translationese," in Findings of ACL 2023, pp. 3614-3633, 2023, doi: 10.18653/v1/2023.findings-acl.223.

[14] S. Mesham, C. Bryant, M. Rei, and Z. Yuan, "An Extended Sequence Tagging Vocabulary for Grammatical Error Correction," in Findings of EACL 2023, pp. 1608-1619, 2023, doi: 10.18653/v1/2023.findings-eacl.119.

[15] H. Cao, Z. Cao, C. Hu, B. Hou, T. Xiao, and J. Zhu, "Improving Autoregressive Grammatical Error Correction with Non-autoregressive Models," in Findings of ACL 2023, pp. 12014-12027, 2023, doi: 10.18653/v1/2023.findings-acl.760.

[16] A. Luhtaru, E. Korotkova, and M. Fishel, "No Error Left Behind: Multilingual Grammatical Error Correction with Pre-trained Translation Models," in Proc. EACL 2024, pp. 1209-1222, 2024, doi: 10.18653/v1/2024.eacl-long.73.

[17] M. R. Qorib, A. F. Aji, and H. T. Ng, "Efficient and Interpretable Grammatical Error Correction with Mixture of Experts," in Findings of EMNLP 2024, pp. 17127-17138, 2024, doi: 10.18653/v1/2024.findings-emnlp.997.

[18] Y. Fei, L. Cui, S. Yang, W. Lam, Z. Lan, and S. Shi, "Enhancing Grammatical Error Correction Systems with Explanations," in Proc. ACL 2023, pp. 7489-7501, 2023, doi: 10.18653/v1/2023.acl-long.413.

[19] Y. Song, K. Krishna, R. Bhatt, K. Gimpel, and M. Iyyer, "GEE! Grammar Error Explanation with Large Language Models," in Findings of NAACL 2024, pp. 754-781, 2024, doi: 10.18653/v1/2024.findings-naacl.49.

[20] G. Peng, W. Li, W. Luo, and H. Wang, "Encode Errors: Representational Retrieval of In-Context Demonstrations for Multilingual Grammatical Error Correction," in Findings of ACL 2025, pp. 21166-21180, 2025, doi: 10.18653/v1/2025.findings-acl.1090.

[21] W. Li, W. Luo, G. Peng, and H. Wang, "Explanation-Based In-Context Demonstrations Retrieval for Multilingual Grammatical Error Correction," in Proc. NAACL 2025, pp. 4881-4897, 2025, doi: 10.18653/v1/2025.naacl-long.251.

[22] P. Bhattacharyya and A. Bhattacharya, "Leveraging LLMs for Bangla Grammar Error Correction: Error Categorization, Synthetic Data, and Model Evaluation," in Findings of ACL 2025, pp. 8220-8239, 2025, doi: 10.18653/v1/2025.findings-acl.431.

[23] V. Dobariya, J. Baxi, B. Gambhava, and B. Bhatt, "Smruti: Grammatical Error Correction for Gujarati using LLMs with Non-Parametric Memory," in Findings of IJCNLP-AACL 2025, pp. 473-485, 2025, doi: 10.18653/v1/2025.findings-ijcnlp.28.

[24] M. Dhamecha, S. Jaat, G. Damor, and P. Mishra, "Team Horizon at BHASHA Task 1: Multilingual IndicGEC with Transformer-based Grammatical Error Correction Models," in Proc. BHASHA 2025, pp. 142-146, 2025, doi: 10.18653/v1/2025.bhasha-1.14.

[25] X. Chang and J. Zhu, "A Chunk-based Chain of Thought Prompting Method for Mitigating Over-Correction in Chinese Grammatical Error Correction," in Proc. CCL 2025, pp. 831-841, 2025.

[26] Y. Wang, B. Wang, Y. Liu, D. Wu, and W. Che, "LM-Combiner: A Contextual Rewriting Model for Chinese Grammatical Error Correction," in Proc. LREC-COLING 2024, pp. 10675-10685, 2024.

[27] X. Liu, B. Xu, M. Yang, H. Cao, C. Zhu, T. Zhao, and W. Lu, "A Chain-of-Task Framework for Instruction Tuning of LLMs Based on Chinese Grammatical Error Correction," in Proc. COLING 2025, pp. 8623-8639, 2025.

[28] X. Wang, L. Mu, J. Zhang, and H. Xu, "Multi-pass Decoding for Grammatical Error Correction," in Proc. EMNLP 2024, pp. 9904-9916, 2024, doi: 10.18653/v1/2024.emnlp-main.553.

[29] T. Goto, Y. Sakai, and T. Watanabe, "gec-metrics: A Unified Library for Grammatical Error Correction Evaluation," in Proc. ACL 2025: System Demonstrations, pp. 524-534, 2025, doi: 10.18653/v1/2025.acl-demo.50.

[30] Y. Sakai, T. Goto, and T. Watanabe, "IMPARA-GED: Grammatical Error Detection is Boosting Reference-free Grammatical Error Quality Estimator," in Findings of ACL 2025, pp. 25647-25654, 2025, doi: 10.18653/v1/2025.findings-acl.1315.

[31] T. Goto, Y. Sakai, and T. Watanabe, "Reliability Crisis of Reference-free Metrics for Grammatical Error Correction," in Findings of EMNLP 2025, pp. 24913-24926, 2025, doi: 10.18653/v1/2025.findings-emnlp.1356.

[32] N. Varshney, S. Mishra, and C. Baral, "Towards Improving Selective Prediction Ability of NLP Systems," in Proc. 7th Workshop on Representation Learning for NLP, pp. 221-226, 2022, doi: 10.18653/v1/2022.repl4nlp-1.23.

[33] P. Zablotskaia, D. Phan, J. Maynez, S. Narayan, J. Ren, and J. Liu, "On Uncertainty Calibration and Selective Generation in Probabilistic Neural Summarization: A Benchmark Study," in Findings of EMNLP 2023, pp. 2980-2992, 2023, doi: 10.18653/v1/2023.findings-emnlp.197.

[34] Y. Ba, M. V. Mancenido, and R. Pan, "Fill In The Gaps: Model Calibration and Generalization with Synthetic Data," in Proc. EMNLP 2024, pp. 17211-17225, 2024, doi: 10.18653/v1/2024.emnlp-main.955.

[35] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. ICML, vol. 70, pp. 1321-1330, 2017.

[36] B. Wilie, et al., "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in Proc. AACL-IJCNLP 2020, pp. 843-857, 2020, doi: 10.18653/v1/2020.aacl-main.85.

[37] S. Cahyawijaya, et al., "IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation," in Proc. EMNLP 2021, pp. 8875-8898, 2021, doi: 10.18653/v1/2021.emnlp-main.699.

[38] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in Proc. EMNLP 2021, pp. 10660-10668, 2021, doi: 10.18653/v1/2021.emnlp-main.833.

[39] R. Mahendra, A. F. Aji, S. Louvan, F. Rahman, and C. Vania, "IndoNLI: A Natural Language Inference Dataset for Indonesian," in Proc. EMNLP 2021, pp. 10511-10527, 2021, doi: 10.18653/v1/2021.emnlp-main.821.

[40] S. Cahyawijaya, et al., "Cendol: Open Instruction-tuned Generative Large Language Models for Indonesian Languages," in Proc. ACL 2024, pp. 14899-14914, 2024, doi: 10.18653/v1/2024.acl-long.796.

[41] T. M. Fahrudin, I. Sa'diyah, Latipah, I. Z. A. Illah, C. C. B. Lirna, and B. S. Acarya, "KEBI 1.0: Indonesian Spelling Error Detection System for Scientific Papers Using Dictionary Lookup and Peter Norvig Spelling Corrector," Lontar Komputer, vol. 12, no. 2, pp. 78-90, 2021, doi: 10.24843/LKJITI.2021.v12.i02.p02.

[42] N. Mediyawati, J. C. Young, and S. B. Nusantara, "U-Tapis: Automatic Spelling Filter as an Effort to Improve Indonesian Language Competencies of Journalistic Students," Cakrawala Pendidikan, vol. 40, no. 2, pp. 402-412, 2021, doi: 10.21831/cp.v40i2.34546.

[43] M. A. Mohsen, "Computer-Mediated Corrective Feedback to Improve L2 Writing Skills: A Meta-Analysis," Journal of Educational Computing Research, vol. 60, no. 5, pp. 1253-1276, 2022, doi: 10.1177/07356331211064066.

[44] N. Zhai and X. Ma, "The Effectiveness of Automated Writing Evaluation on Writing Quality: A Meta-Analysis," Journal of Educational Computing Research, vol. 61, no. 4, pp. 875-900, 2023, doi: 10.1177/07356331221127300.

[45] N. Fan, "Exploring the Effects of Automated Written Corrective Feedback on EFL Students' Writing Quality: A Mixed-Methods Study," SAGE Open, vol. 13, no. 2, 2023, doi: 10.1177/21582440231181296.

[46] S. S. Lee and R. L. Moore, "Harnessing Generative AI (GenAI) for Automated Feedback in Higher Education: A Systematic Review," Online Learning, vol. 28, no. 3, 2024, doi: 10.24059/olj.v28i3.4593.

[47] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in Proc. COLING, pp. 757–770, 2020, doi: 10.18653/v1/2020.coling-main.66.

[48] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi, "Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics," in Proc. EMNLP 2020, pp. 9275-9293, 2020, doi: 10.18653/v1/2020.emnlp-main.746.

[49] C. G. Northcutt, L. Jiang, and I. L. Chuang, "Confident Learning: Estimating Uncertainty in Dataset Labels," Journal of Artificial Intelligence Research, vol. 70, pp. 1373-1411, 2021, doi: 10.1613/jair.1.12125.

[50] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. Aroyo, "Everyone Wants to Do the Model Work, Not the Data Work: Data Cascades in High-Stakes AI," in Proc. CHI 2021, art. 39, 2021, doi: 10.1145/3411764.3445518.

[51] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," Soviet Physics Doklady, vol. 10, no. 8, pp. 707-710, 1966.

[52] E. B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference," Journal of the American Statistical Association, vol. 22, no. 158, pp. 209-212, 1927.

[53] Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," Psychometrika, vol. 12, pp. 153-157, 1947.

[54] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap. New York, NY, USA: Chapman & Hall, 1993.

CompCap-Stego: Computable Capability-Pointer Steganography with Hidden Policy Evaluation and Verifiable External Payload Retrieval

Mohammad Othman Nassar
Faculty of Information Technology-Cyber Security Department, Amman Arab University, Amman, Jordan

***Abstract*—CompCap-Stego is a capability-pointer steganography framework in which the concealed object is a compact dual-zone capability capsule rather than the protected payload. The sealed access zone stores receiver binding, wrapped key material, a revocation handle, an audit nonce, and a payload-verification root, while the computable policy zone supports validity, expiry, bounded-use access, revocation, replay state, receiver matching, honey-state misuse detection, and policy-digest consistency. Here, computable denotes authenticated policy-state evaluation after capsule extraction rather than general-purpose or homomorphic computation directly over unrecovered steganographic bits. Retrieval proceeds only after capsule extraction, binding validation, policy evaluation, receiver-bound token derivation, and verification of the externally returned encrypted payload. A 512-byte core capsule reduces embedded size by 95.00 percent for a 10-kilobyte (KB) payload, 99.50 percent for 100 KB, 99.95 percent for a 1-megabyte (MB) payload, and 99.995 percent for 10 MB. Carrier quality remains stable at 55.8 to 55.9 decibels (dB) in peak signal-to-noise ratio (PSNR), 0.997 structural similarity index measure (SSIM), and 0.56 area under the receiver operating characteristic curve (AUC). Medium error correction reaches 99.6 percent extraction success and 99.5 percent tag-verification success. Policy violations are rejected at 98.9 to 99.8 percent, tampered or replaced external ciphertext at 99.8 to 99.9 percent, and the full design records 0.4 percent composite attack success.**

Keywords*—Steganography; capability-based access control; hidden policy evaluation; verifiable external retrieval

I. Introduction

Steganography has traditionally concealed secret messages in innocuous carriers while maintaining the cover's perceptual or statistical properties. Across image, audio, video, network, linguistic, and generative carriers, this objective continues to create a central tension among capacity, imperceptibility, robustness, and extraction reliability [1]–[7]. More recent work has extended steganography beyond message hiding to security functions that include hidden computation, authentication support, integrity proof transmission, and high-capacity generative embedding [8]–[13]. This progression broadens the role of the hidden object: in addition to a conventional message, it can serve as a control object, proof object, credential object, or retrieval-enabling object.

Full-payload embedding provides a direct route to controlled information access, but it is not always the most suitable option. When the protected payload is large, dynamic, revocable, or externally managed, placing the complete ciphertext in the cover increases capacity pressure and carrier distortion. A smaller hidden object reduces this embedding load, yet a static hidden pointer or static token leaves the access-state weakness unresolved. Once extracted or replayed, it may continue to expose the external payload unless access state, receiver binding, revocation, replay, and payload verification are enforced. The intervention point therefore lies between compact pointer-style hiding and stateful access-controlled retrieval.

Adjacent literature addresses several parts of this problem separately. Homomorphic and computable steganography demonstrate that computation can be applied to concealed bits without directly revealing them [8]. Anonymous-token systems and private-metadata token constructions provide mechanisms for unlinkability, private metadata, public verification, and double-spending control [14]–[18]. Internet of Things (IoT) authorization and device-sharing research shows the risks of exposed authorization data and the need for delegation, revocation, and access-state management [19]–[21]. Privacy-oriented concealment also addresses exposure of access-related network traces [22]. Capability and network-bound access-control systems provide process or context binding [23], while encrypted-domain data-hiding and proxy-re-encryption systems support controlled sharing of protected data [24]–[27]. Nevertheless, these lines do not provide a single steganographic construction in which the concealed object is a compact computable capability capsule that controls verified retrieval of an externally encrypted payload.

Carrier-level developments in image-based information hiding and cover selection provide a second motivation for CompCap-Stego. Research on cover-image selection establishes that carrier suitability affects embedding reliability, visual quality, and resistance to detection, whereas high-capacity generative steganography shows that modern carriers can be constructed or adapted for larger and more robust hidden representations [10]. CompCap-Stego takes a different but complementary direction: instead of increasing the hidden payload placed in the carrier, it confines the hidden object to a compact access capsule and moves the protected content to verified external encrypted storage. The design emphasis therefore shifts from maximizing embedded payload capacity to minimizing embedded access material while retaining policy enforcement, receiver binding, and payload verification.

This study proposes CompCap-Stego, a computable capability-pointer steganography framework for hidden policy evaluation and verifiable external payload retrieval. In this study, computable denotes a limited authenticated state-transition capsule

whose concealed policy fields support receiver-side and gateway-side policy evaluation after extraction; it does not denote general-purpose computation over unrecovered steganographic bits. Rather than embedding the payload, the framework embeds a dual-zone capability capsule. Its sealed access zone protects retrieval material, receiver binding, wrapped key material, payload-verification root, revocation handle, and audit nonce, whereas its computable policy zone maintains compact hidden state for validity, expiry, bounded-use retrieval, revocation, replay, receiver matching, honey-state misuse detection, and policy-digest checking. Retrieval is granted only after the capsule is extracted, its binding is verified, the hidden policy predicate is satisfied, a receiver-bound retrieval token is derived, and the returned external ciphertext matches the verification root bound to the capsule. The principal novelty therefore lies in combining a concealed capability capsule, authenticated stateful policy-before-retrieval control, receiver-bound authorization, and external-payload verification rather than in introducing a new form of homomorphic steganographic computation.

The evaluation demonstrates the role of the capsule-centered design across compactness, carrier quality, extraction reliability, policy enforcement, payload verification, runtime overhead, and ablation behavior. Compared with full-payload embedding, CompCap-Stego substantially reduces the number of embedded bits as payload size increases because the cover carries a capsule of bounded size rather than the full encrypted payload. Compared with static pointer and static sealed-token variants, the proposed method provides replay, revocation, expiry, receiver-mismatch, honey-probing, and payload-replacement rejection through the policy and verification layers.

A. Contributions

CompCap-Stego makes five linked contributions. First, it defines computable capability-capsule steganography in which the hidden object is a compact capability capsule rather than the full payload, a plain pointer, or an ordinary hidden credential. Second, it introduces a dual-zone capsule composed of a sealed access zone for retrieval and verification material and a computable policy zone for compact hidden policy state. Third, it evaluates policy-before-retrieval rules for validity, expiry, bounded-use access, revocation, replay, receiver matching, and honey-state misuse detection. Fourth, it binds external ciphertext acceptance to a capsule-sealed hash or Merkle root to prevent replacement and tampering. Fifth, it compares the design with full-payload embedding, static pointer hiding, static sealed-token hiding, and component-removal variants.

Section II reviews the closest steganography, token, capability, and verifiable-retrieval literature. Section III defines the system and threat models, including the gateway trust boundary. Section IV presents the CompCap-Stego construction, followed by the formal algorithms and security properties in Section V. Section VI describes the experimental methodology and sensitivity analyses, while Section VII reports the quantitative results. Section VIII discusses the resulting design trade-offs, Section IX identifies the validated scope and remaining limitations, and Section X concludes the study.

II. Related Work

This section positions CompCap-Stego relative to six adjacent families of work: conventional and adaptive steganography, computable and generative steganography, anonymous/private-metadata tokens, authorization-data and capability-based access control, steganography-assisted authentication [28], [29], and encrypted-domain retrieval with integrity verification.

A. Steganography, Steganalysis, and Carrier Selection

Classical steganography and modern distortion-minimization methods optimize the capacity-imperceptibility balance [1]–[7]. Adaptive image methods reduce detectable artifacts by selecting low-risk embedding regions, while deep steganalysis models increase the difficulty of remaining undetected. These methods are important for the carrier layer of CompCap-Stego, but they do not define the hidden object as a stateful retrieval capability. StyleStego is related at the carrier layer because it explores high-capacity generative steganography using a style-based generative adversarial network (StyleGAN) embedding [10]. In addition, optimization-assisted color visual cryptography is relevant as adjacent image-security work because it addresses protected visual-content representation using image-domain security mechanisms [11].

B. Computable and Generative Steganography

ProSt shows that homomorphic-style computation can be performed on hidden steganographic data [8]. Generative and neural approaches hide information by synthesizing carriers or learning end-to-end encoder/decoder structures [9], [12], [13]. CompCap-Stego does not claim to be the first work on hidden computation or generative hiding. Its scope is narrower: the hidden computation is limited to policy-state evaluation over a compact capsule, and the protected payload remains external. Unlike systems that operate directly on concealed or unrecovered steganographic data, CompCap-Stego first extracts and authenticates the capsule and then evaluates its protected policy state. The term computable is therefore used in the narrower sense of stateful capability evaluation.

C. Anonymous Tokens and Private Metadata

Anonymous tokens with private metadata bits and public metadata support privacy-preserving issuance, redemption, metadata hiding, public verification, and double-spending controls [14]–[18]. These systems strongly motivate the use of compact credential-like objects, but they are not steganographic carrier objects and do not bind an embedded capsule to off-cover ciphertext retrieval. CompCap-Stego combines the token-like idea of a compact access object with carrier concealment and payload-root verification.

D. Authorization Data, Capability Systems, and External Access Control

IoT device-sharing and authorization-data studies identify the underlying access-control weakness: exposed access material can be replayed or remain usable after revocation [19]–[21]. Privacy-oriented concealment work addresses a related exposure by hiding network-use traces through stealth protocols and proxying when access activity can reveal sensitive behavior [22]. Capability and network-bound systems such as NetCap respond to token theft by binding access authority to a process or network context [23]. These mechanisms provide access control, but they do not operate as steganographic retrieval capsules. CompCap-Stego retains the context-binding function and places the resulting authority inside a concealed carrier capsule.

E. Encrypted-Domain Data Hiding and Verifiable Retrieval

Reversible data hiding in encrypted domains and proxy re-encryption-based data hiding provide controlled sharing, identity verification, and encrypted-domain embedding [24]–[27]. Merkle-based integrity verification and steganographic proof transmission add the complementary function of tamper-evident verification to hidden transmission [30], [31]. These approaches therefore cover part of the proposed framework's verification role, but not the dual-zone hidden capability capsule that evaluates policy before retrieval. The functional boundaries of the closest work families are summarized in Table I.

TABLE I. FUNCTIONAL COMPARISON WITH CLOSEST WORK FAMILIES

Work family	Representative focus and citations	Overlap with CompCap-Stego	Remaining gap and claim boundary
Adaptive and deep steganography	Capacity, imperceptibility, robustness, and steganalysis resistance [1]-[7], [10]-[13]	Provides carrier-layer methods and detectability metrics.	Does not provide stateful external retrieval authority; CompCap-Stego does not claim novelty in adaptive or generative embedding.
Computable steganography	Computation over hidden steganographic data [8]	Supports the idea that hidden bits can participate in computation.	Does not define a compact access capsule for verified retrieval; CompCap-Stego is not claimed as the first hidden-computation method.
Anonymous and private-metadata tokens	Private metadata, public metadata, verification, and double-spending control [14]-[18]	Supports compact token-like access objects and replay reasoning.	No steganographic carrier binding or external ciphertext root; CompCap-Stego is not claimed as the first anonymous-token system.
Authorization-data protection and capability systems	Delegation, revocation, hidden authorization data, and network/process-bound capabilities [19]-[23]	Motivates revocation, receiver binding, and access-state management.	Does not place a computable retrieval capsule inside a cover; CompCap-Stego is not claimed as the first access-control or capability system.
Encrypted-domain data hiding and proxy re-encryption	Controlled sharing, encrypted-domain embedding, and dynamic authorization [24]-[27]	Overlaps with external encrypted data and controlled retrieval.	No policy-before-retrieval capsule in the carrier; CompCap-Stego is not claimed as the first encrypted-domain hiding method.
Integrity-proof steganography and Merkle verification	Merkle or proof transmission through covert channels [30], [31]	Overlaps with tamper-evident proof handling.	Does not combine sealed retrieval, hidden policy, and receiver-bound access; CompCap-Stego is not claimed as the first Merkle-verification method.

Accordingly, the claim boundary is deliberately limited: CompCap-Stego is not positioned as the first hidden-computation, anonymous-token, access-control, Merkle-verification, or encrypted-domain data-hiding method. Its contribution rests on the unified capsule-level mechanism that combines a sealed access zone, a computable policy zone, honey positions, and verified external retrieval inside a concealed carrier object.

III. SYSTEM MODEL AND THREAT MODEL

A. System Overview

CompCap-Stego considers a sender that protects a payload by encrypting it externally and embedding a compact capability capsule in a cover object. The receiver extracts the capsule, validates its binding, evaluates hidden policy state, derives a retrieval token, and accepts the externally returned ciphertext only if it matches the verification root carried in the capsule.

$$\Gamma = Z_s \parallel Z_p \parallel \tau \tag{1}$$

where, Γ denotes the embedded capability capsule, Z_s is the sealed access zone, Z_p is the computable policy zone, and τ is the capsule binding tag.

$$C' = Embed(C, \Gamma, K_{emb}) \tag{2}$$

where, C is the original cover, C' is the stego-cover, and K_{emb} is the embedding key.

$$M = (R, D, H) \tag{3}$$

The position map M separates carrier positions into real positions R, decoy positions D, and honey positions H.

$$R \cap D = R \cap H = D \cap H = \varnothing \tag{4}$$

B. External Payload and Policy State

The protected content is encrypted before storage using authenticated encryption with associated data (AEAD), where AD denotes the associated data. The receiver does not accept the retrieved ciphertext merely because it is returned from the external store; acceptance depends on the hash or Merkle-root verification material bound to the capsule [32]. The core notation is summarized in Table II.

TABLE II. CORE NOTATION

Symbol	Meaning
Γ	Complete embedded capability capsule
Z_s	Sealed access zone
Z_p	Computable policy zone
τ	Capsule binding tag
C, C'	Cover and stego-cover
P, P_e	Plain payload and encrypted payload
$root_P$	Hash or Merkle root for external payload verification
R, D, H	Real, decoy, and honey position classes
R_ctx	Receiver context used in receiver-binding and policy evaluation.
c_u	Remaining-use counter
b_{rep}, b_r, b_h	Replay, revocation, and honey-state flags

$$P_e = AEAD.Enc_{K_e}(P, AD) \tag{5}$$

$$root_P = H(P_e) \tag{6}$$

$$root_P = MerkleRoot(ch_1, ch_2, \dots, ch_n) \tag{7}$$

$$Z_p = \left(b_v, b_x, c_u, b_r, b_{rep}, b_{rec}, b_h, d_p\right) \tag{8}$$

$$Allow(\Gamma, R, t) = b_v \wedge \neg b_x \wedge (c_u > 0) \wedge \neg b_r \wedge \neg b_{rep} \wedge b_{rec} \wedge \neg b_h \tag{9}$$

C. Threat Model and Security Goals

The threat model allows the adversary to observe stego-covers, attempt extraction, replay a previously extracted capsule, request retrieval through an unauthorized receiver context, trigger decoy or honey positions, modify the stego-cover, tamper with the external ciphertext, substitute another payload, or attempt to bypass revocation under an active-network adversary abstraction [33]. The adversary is assumed not to know the embedding key, sealing key, tag key, or receiver secret material. Within this boundary, the security goals comprise capsule confidentiality, extraction correctness for authorized receivers, replay rejection, revocation enforcement, receiver binding, honey misuse rejection, external payload integrity, and compactness relative to full-payload embedding. The external payload store is not trusted for ciphertext integrity, whereas the gateway is trusted to preserve the correctness and monotonicity of mutable policy state. Receiver binding addresses unauthorized receiver contexts rather than complete compromise of an authorized receiver secret. Replay, revocation, and bounded-use guarantees therefore depend on the integrity of gateway state; malicious gateway compromise or state rollback falls outside these guarantees.

D. Gateway Trust, State Consistency, and Concurrency Assumptions

Replay and bounded-use enforcement require atomic state transitions keyed by the capsule identifier and audit state. Concurrent requests for the same capsule are therefore assumed to be serialized or resolved through an atomic compare-and-update operation so that only a state-consistent transition can authorize retrieval. Reuse of a consumed retrieval token is rejected while the corresponding gateway state remains intact.

These guarantees do not extend to a malicious or rolled-back gateway state. Restoring an earlier counter, replay flag, or revocation record can invalidate replay, revocation, and bounded-use enforcement even when the capsule remains cryptographically intact. The present security boundary therefore assumes correct gateway state maintenance and does not claim replicated-gateway consensus or resilience to full compromise of an authorized receiver secret.

IV. CompCap-Stego Design

The framework connects five layers: payload protection, capability-capsule construction, hidden policy computation, steganographic embedding, and retrieval verification. Their combined function is to conceal a compact access object in the carrier while preserving the cryptographic binding of the externally stored ciphertext to the capsule. The overall architecture is shown in Fig. 1.

A. Dual-Zone Capsule

The sealed access zone contains retrieval and verification material that should not be exposed by extraction alone. The computable policy zone contains compact state bits that can be evaluated before retrieval. The message authentication code (MAC) tag binds both zones to the cover context, salt, and suite version. The capsule layout is shown in Fig. 2.

$$\Gamma = Z_s \parallel Z_p \parallel \tau \tag{10}$$

$$Z_s = Seal_{K_{seal}}\left(pid, sid, rid_h, store_{ref}, K_w, root_P, rev_h, n_a, v\right) \tag{11}$$

$$Z_p = \left(b_v, b_x, c_u, b_r, b_{rep}, b_{rec}, b_h, d_p\right) \tag{12}$$

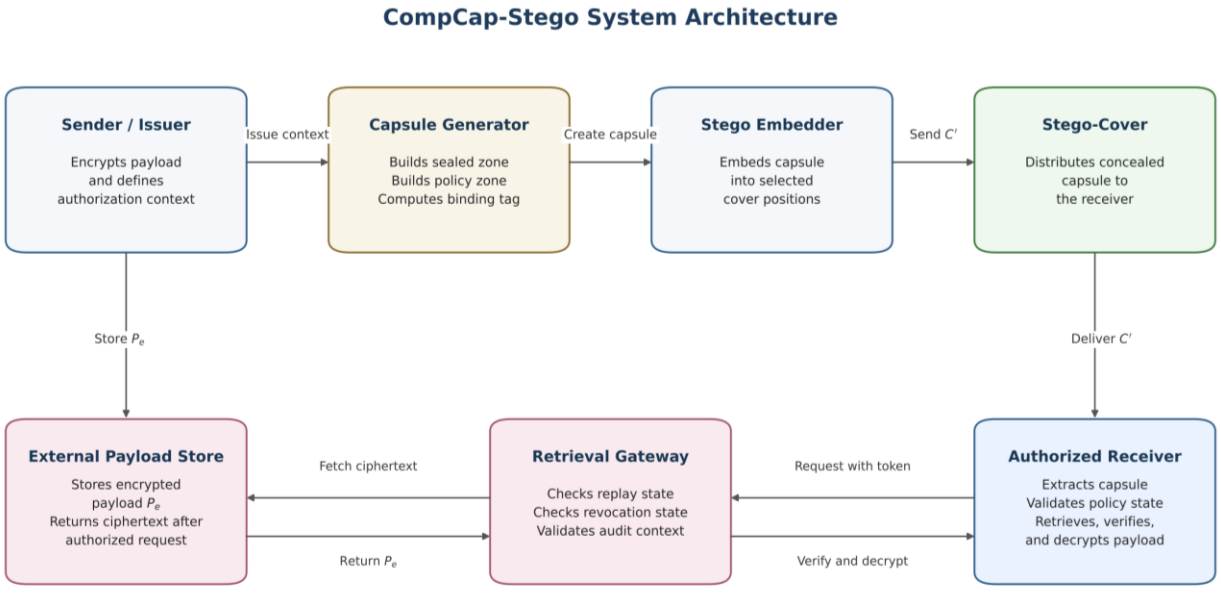


Fig. 1. Architecture of the CompCap-Stego showing capsule generation, steganographic embedding, hidden policy evaluation, gateway mediation, external payload retrieval, and audit logging.

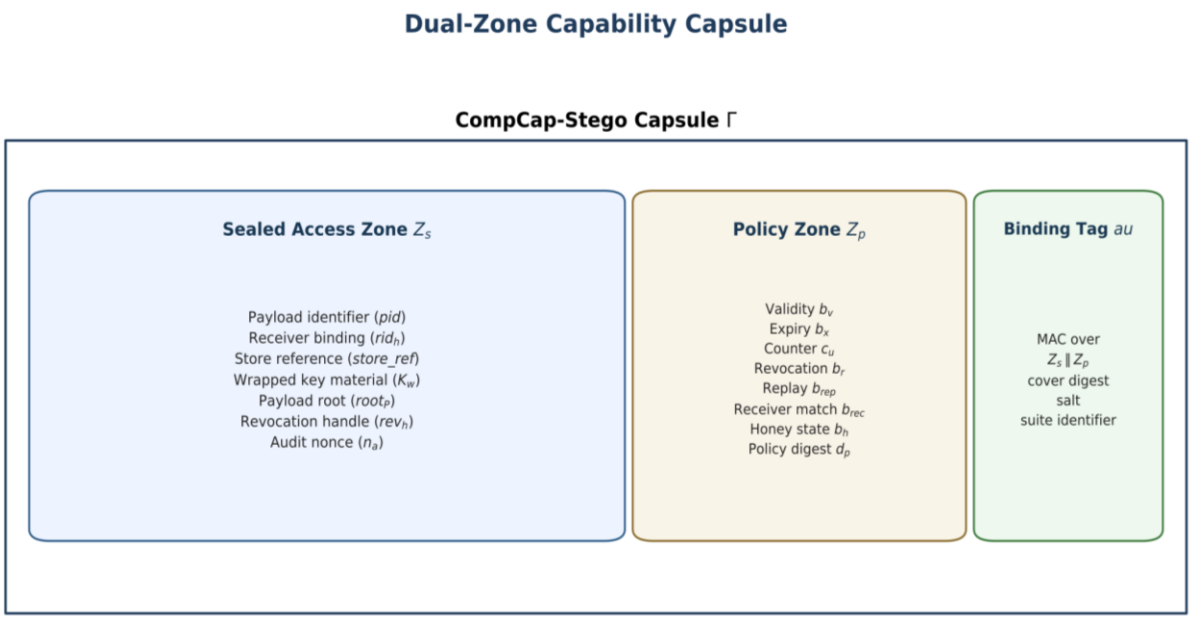


Fig. 2. Dual-zone capsule structure with sealed access fields, computable policy fields, and the binding tag.

B. Binding, Retrieval Token, and Position Classes

The capsule tag authenticates the two capsule zones against the cover context and suite version. The retrieval token is receiver-bound and depends on the payload identifier, receiver hash, revocation handle, audit nonce, policy state, and receiver secret.

The real positions carry the true capsule, decoy positions carry plausible false bits, and honey positions support misuse detection. All position classes are processed uniformly enough to make probing more difficult, while only the authenticated capsule path authorizes retrieval. The position classes are shown in Fig. 3. The complete policy-before-retrieval sequence is shown in Fig. 4.

$$\tau = MAC_{K_{tag}}\left(Z_s \parallel Z_p \parallel H(C) \parallel salt \parallel v\right) \quad (13)$$

$$Allow(\Gamma, R_{ctx}, t) = b_v(t) \wedge \neg b_x(t) \wedge [c_u(t) > 0] \wedge \neg b_r(t) \wedge \neg b_{rep}(t) \wedge b_{rec}(R_{ctx}) \wedge \neg b_h(t) \quad (14)$$

$$rtok = H\left(pid \parallel rid_h \parallel rev_h \parallel n_a \parallel Z_p \parallel sk_R\right) \quad (15)$$

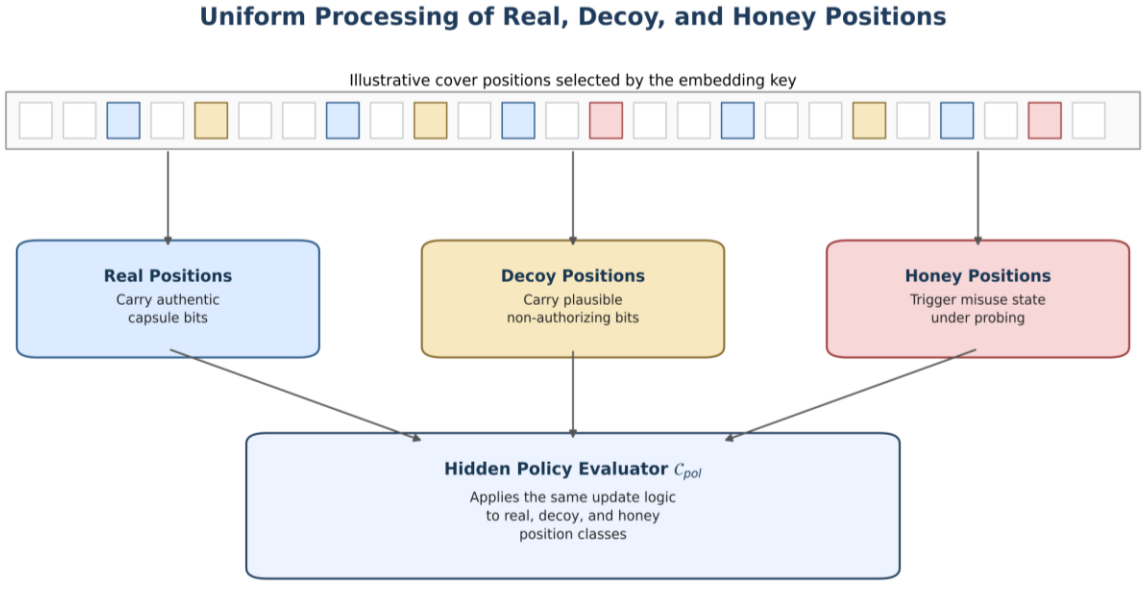


Fig. 3. Real, decoy, and honey position classes used by the embedding and misuse-detection layers.

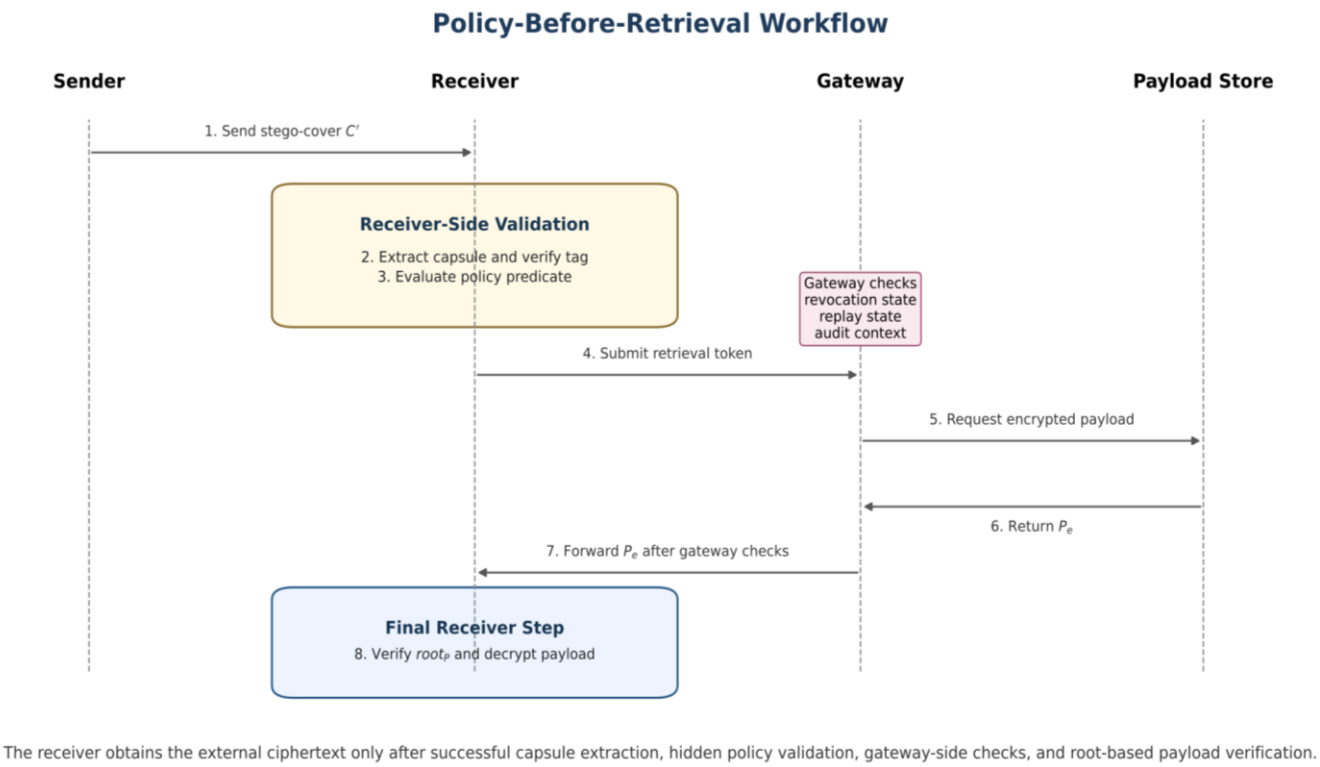


Fig. 4. Policy-before-retrieval workflow from extraction through gateway-mediated payload verification.

V. Formal Algorithms and Security Analysis

A. State Transition and Payload Verification

Let the policy zone at the retrieval event t be denoted by Z_p^t, and let e_t denote the observed event. The hidden policy evaluator maps the current policy state and event to an updated policy state. The updated state is then used for policy-before-retrieval authorization and external ciphertext verification.

$$Z_p^{t+1} = C_{pol}\left(Z_p^t, e_t\right) \tag{16}$$

$$c_u^{t+1} = max(c_u^t - 1{,}0) \tag{17}$$

$$b_{rep}^{t+1} = b_{rep}^t \vee \delta_{rep,t} \tag{18}$$

$$b_r^{t+1} = b_r^t \vee \delta_{rev,t} \tag{19}$$

$$b_h^{t+1} = b_h^t \vee \delta_{h,t} \tag{20}$$

$$d_p^{t+1} = H\left(b_{pol}^{t+1}\right) \tag{21}$$

$$A_e = 1 \Leftrightarrow H(P_e') = r_P \tag{22}$$

$$A_i = 1 \Leftrightarrow V_M(H(ch_i'), \pi_i, r_P) = 1 \tag{23}$$

$$P = \begin{cases} Dec_{K_e}(P_e', AD), & A_e = 1, \\ \perp, & A_e = 0. \end{cases} \tag{24}$$

The indicator terms $\delta_{rep,t}$, $\delta_{rev,t}$, and $\delta_{h,t}$ represent replay or consumption, revocation, and honey-trigger events, respectively; b_{pol}^{t+1} denotes the ordered policy-bit vector after the state update. The terms A_e and A_i denote object-level and chunk-level acceptance, r_P denotes the payload verification root, and V_M denotes Merkle-path verification.

B. Algorithms

The five procedures used by the framework are summarized in Table III.

TABLE III. Algorithmic Procedures

Algorithm	Input	Main output	Role
Algorithm 1: CapsuleGen	Payload, cover, receiver context, policy, keys	Stego-cover and external ciphertext record	Encrypts payload, constructs dual-zone capsule, binds the capsule, and embeds it.
Algorithm 2: HiddenPolicyEval	Policy state, event, evaluator, position context	Updated policy state	Updates expiry, revocation, replay, receiver-match, honey, and use-counter state.
Algorithm 3: CapsuleExtract	Stego-cover, receiver credentials, embedding and tag keys	Validated capsule or rejection	Extracts, parses, authenticates, opens, and validates the capsule.
Algorithm 4: PayloadRetrieveVerify	Validated capsule, receiver secret, gateway, store	Plaintext payload or rejection	Derives retrieval token, mediates gateway retrieval, verifies ciphertext, and decrypts.
Algorithm 5: AttackEval	Capsule, ciphertext, attack type, configuration	Attack outcome and metrics	Evaluates replay, revocation bypass, expiry bypass, receiver mismatch, honey probing, tampering, and replacement.

The five procedures in Table III were instantiated as executable pseudocode as follows. The notation is deliberately implementation-oriented so that the generation, extraction, policy, retrieval, and attack-evaluation paths can be reproduced from the same inputs.

Algorithm 1: CapsuleGen

Input: payload, cover image, receiver context, policy fields, embedding key, sealing key, tag key.
Output: stego-cover, external ciphertext record, and audit seed.
Steps:
1. Canonize the cover context, receiver fingerprint, suite identifier, and policy fields.
2. Encrypt the payload using the Advanced Encryption Standard with a 256-bit key in Galois/Counter Mode (AES-256-GCM) [34], [35] and calculate the payload hash or Merkle root.
3. Build the sealed access zone from the payload identifier, wrapped key material, receiver hash, revocation handle, audit nonce, and verification root.
4. Encode the policy zone with the masked validity, expiry, bounded-use, replay, revocation, receiver-match, honey-state, and policy-digest fields.
5. Authenticate both zones using the hash-based message authentication code with the 256-bit Secure Hash Algorithm (HMAC-SHA-256) [36] and apply the selected Reed–Solomon profile.
6. Generate the keyed real/decoy/honey position map and embed the authenticated capsule into the cover image.
7. Return the stego-cover, external ciphertext record, and audit seed.

Algorithm 2: HiddenPolicyEval

Input: extracted capsule state, receiver context, gateway state, retrieval event.
Output: updated policy decision and updated audit state.
Steps:
1. Validate the capsule's authenticity and recover the masked policy-state vector.
2. Check the policy state for validity, expiry, revocation handle, bounded-use count, receiver match, replay flag, and honey-state indicator.
3. Mark the event as allow if and only if all policy predicates are met.
4. If any of the predicates are not met, mark the event as reject and record the reason for the rejection.
5. Update the gateway's counters, replay marker, and audit chain for the capsule identifier and audit nonce.
6. Return the policy decision and the synchronized policy and audit state.

Algorithm 3: CapsuleExtract

Input: stego-cover, embedding key, receiver credentials, tag key.
Output: validated capsule or rejection.
Steps:
1. Regenerate the keyed candidate-position map for the stego-cover.
2. Extract the bit streams from the real, decoy, and honey candidate classes of the stego-cover using the selected embedding profile.
3. Apply the configured error-correction decoder to each of the candidate capsule streams.
4. Parse each capsule stream to determine the sealed access zone, the policy zone, and the binding tag.
5. Reject each candidate capsule stream that fails validation of its length, version, profile, digest, or HMAC.
6. If exactly one of the candidate capsule streams satisfies all of the capsule checks, then return that validated capsule. Otherwise, return rejection.

Algorithm 4: PayloadRetrieveVerify

Input: validated capsule, receiver secret, gateway interface, external store.
Output: plaintext payload or rejection.
Steps:
1. Derive the receiver-bound retrieval token from the receiver secret, payload identifier, receiver hash, revocation handle, audit nonce, and policy digest contained within the capsule.
2. Submit this retrieval token and the capsule identifier to the gateway.
3. Before releasing any external ciphertext to the receiver, apply HiddenPolicyEval.
4. Retrieve the encrypted payload from the gateway only if it returns an allow decision.
5. Verify the retrieved encrypted payload by comparing it against the sealed hash or Merkle root within the capsule.
6. If the encrypted payload matches the sealed hash or Merkle root, decrypt it using the wrapped key material from the capsule.
7. Return the plaintext payload or reject the retrieval attempt.

Algorithm 5: AttackEval

Input: configuration, attack scenario, carrier set, payload set, seeds.
Output: rejection rates, false-allow rates, runtime, and composite attack success.
Steps:
1. Generate stego-covers and external ciphertext records for each of the provided seeds and the carrier subset of those seeds.
2. Apply the selected attack scenario to each of the generated stego-covers. The attack scenarios included were replay, revoked state, expiry, wrong receiver, honey probing, ciphertext tampering, payload replacement, chunk tampering, or missing chunk.
3. Run the extraction, policy evaluation, gateway mediation, and payload verification algorithms for each of the generated attack traces.
4. Record the allow, reject, false allow, false reject, verification failure, and latency outcomes for each attack trace.
5. Calculate the rejection and false-allow rates for each of the selected attack scenarios and then calculate the composite attack success according to Eq. (32).
6. Return the table containing the computed rates for each attack scenario as well as the source values used to generate the figure plots that corresponded to these computed rates.

C. Security Arguments

The following security arguments are stated under the assumption that the embedding key, sealing key, authentication key, and receiver secret material are unavailable to the adversary, and that AES-256-GCM, HMAC-SHA-256, the HMAC-based Extract-and-Expand Key Derivation Function using SHA-256 (HKDF-SHA-256), and the selected hash or Merkle commitment are used according to their standard security requirements. The analysis is property-based rather than a full reductionist cryptographic proof, and the adversary is modeled through extraction, replay, revocation-bypass, receiver-mismatch, honey-probing, ciphertext-tampering, replacement, and missing-chunk queries.

$$Pr\left[\hat{\Gamma} = \Gamma\right] \geq 1 - \varepsilon_{ext} \quad (25)$$

$$Pr\left[H(P_e\ ) = H(P_e) \wedge P_e\ \neq P_e\right] \leq negl(\lambda) \quad (26)$$

$$b_{rep} = 1 \Rightarrow Allow(\Gamma, R, t) = 0 \quad (27)$$

$$b_r = 1 \Rightarrow Allow(\Gamma, R, t) = 0 \quad (28)$$

$$\chi'_R \neq rid_h \Rightarrow rtok = \perp \quad (29)$$

$$b_h = 1 \Rightarrow Allow(\Gamma, R, t) = 0 \quad (30)$$

$$|\Gamma| < |P_e| \Rightarrow \rho = 1 - |\Gamma|/|P_e| > 0 \quad (31)$$

Eq. (25) states capsule correctness under the configured channel and error-correction assumptions. Eq. (26) states payload-binding integrity under collision resistance. Eq. (27)–(30) state replay, revocation, receiver-binding, and honey-state rejection, where denotes the challenged receiver-binding value. Eq. (31) states positive embedded-size reduction when the capsule is smaller than the encrypted payload. Table IV summarizes these security properties.

TABLE IV. SECURITY ANALYSIS SUMMARY

Property	Analysis statement
Capsule correctness	Valid keys and acceptable carrier distortion allow the receiver to recover the generated capsule.
Tag and digest validation	Capsule-state modification, field swapping, and inconsistent policy updates are rejected by binding and digest checks.
Replay rejection	A consumed or replayed capsule fails the policy predicate and gateway audit check.
Revocation rejection	A capsule associated with an active revocation state fails retrieval authorization.
Receiver binding	A receiver mismatch prevents valid retrieval-token derivation.
Honey misuse handling	Misuse or probing of honey positions shifts the policy state into rejection.
Payload verification	External ciphertext is accepted only when hash or Merkle verification matches the sealed payload root.
Compactness	The cover embeds the capsule rather than the full encrypted payload.

VI. EXPERIMENTAL METHODOLOGY

The experimental comparison is structured around three baselines and component-removal variants of CompCap-Stego. The baseline comparisons isolate the effect of replacing full-payload embedding with a compact capability capsule, while the ablation variants isolate the contribution of policy state, payload binding, replay state, receiver binding, and honey positions. The experiments used a reproducible image-carrier protocol, explicit cryptographic parameterization, repeated runs, and statistical uncertainty reporting so that compactness, image quality, steganalysis exposure, extraction reliability, policy enforcement, payload verification, runtime, and ablation behavior can be compared under the same carrier and attack conditions.

A. Carrier Corpus, Payload Generation, and Experimental Repetitions

The carrier corpus contained 1,200 grayscale natural images sampled without replacement from BOSSBase 1.01, a public steganographic-image benchmark of 512 × 512 grayscale natural covers [37]. The sampled covers retained the original grayscale content and were stored in Portable Network Graphics (PNG) format after lossless conversion from the source cover format. Images were selected across the benchmark index range to avoid adjacent-image clustering and were partitioned into 800 training covers, 200 validation covers, and 200 test covers for steganalysis and carrier-quality evaluation. No test image was used for embedding-parameter selection. Four external encrypted-payload sizes were evaluated: 10 kilobytes (KB), 100 KB, 1 megabyte (MB), and 10 MB. CompCap-Stego embedded only the fixed-size capsule, whereas the full-payload baseline attempted to embed the complete encrypted payload. Each reported configuration was evaluated over ten deterministic seeds, with 120 covers sampled per seed. The 12,000 carrier-level observations were computed as 120 covers × 10 seeds × 10 embedding/capsule configurations, and policy-violation outcomes used 1,000 attempts × 10 seeds, yielding 10,000 decision observations per policy-violation condition.

B. Embedding, Position Classes, and Error Correction

The embedding layer used an adaptive spatial embedding schedule driven by a keyed pseudo-random position map. Candidate locations were partitioned into real, decoy, and honey classes using a 1:2:1 allocation over the selected embedding pool. Real positions carried the authenticated capsule bits, decoy positions carried plausible capsule-like redundancy that failed authentication, and honey positions carried markers that intentionally activated misuse-state handling when probed. The core capsule was 512 bytes. The light error-correction coding (ECC) profile expanded the embedded object to 640 bytes using 128 bytes of Reed–Solomon parity over byte blocks, while the medium profile expanded it to 768 bytes using 256 bytes of parity. These profiles correspond to the reliability values reported in Table IX.

1) Carrier embedding implementation: The underlying embedding primitive was spatial-domain adaptive ±1 least-significant-bit (LSB) matching with syndrome-trellis coding over a keyed candidate-location sequence, following distortion-minimization practice [2], [3]. For each 512 × 512 cover, admissible pixels were ranked by a local high-pass residual and variance score; saturated pixels and boundary values were excluded. The protected capsule bitstream was embedded at 0.0156 bits per pixel (bpp) for the 512-byte core, 0.0195 bpp for the 640-byte light-ECC profile, and 0.0234 bpp for the 768-byte medium-ECC profile. Real positions carried authenticated capsule symbols, while decoy and honey positions were physically populated with keyed filler symbols drawn from the same admissible carrier-position pool. Thus, extraction probes encountered plausible streams, but only the authenticated real stream opened under the capsule tag. Covers below the required low-distortion capacity threshold for a configuration were excluded from that configuration rather than padded with high-distortion changes.

C. Cryptographic Instantiation and Receiver Binding

The sealed access zone used AES-256-GCM with a 96-bit nonce and a 128-bit authentication tag. Per-capsule keys were derived using HKDF-SHA-256 from an issuer secret, a capsule salt, the receiver fingerprint, and the suite identifier. Capsule

authentication used HMAC-SHA-256 over the sealed access zone, the computable policy zone, the cover-context digest, and the suite version. Receiver binding was computed from the receiver fingerprint, retrieval context, revocation handle, and audit nonce; retrieval-token derivation failed when these fields did not match the authorized receiver context. External ciphertext acceptance used either a SHA-256 payload hash or a Merkle root sealed inside the capsule.

D. Computable Policy-Zone Instantiation

The computable policy zone was represented as a compact masked state vector containing validity, expiry, bounded-use counter, revocation, replay, receiver-match, honey-state, and policy-digest fields. In this study, computable policy evaluation is a constrained state-transition mechanism over authenticated capsule fields, not arbitrary computation over unrecovered steganographic content. The receiver first recovered and authenticated the capsule, then evaluated the masked policy vector through the same keyed position map used to distinguish real, decoy, and honey classes. The evaluation was therefore concealed at the carrier-position level rather than performed as general-purpose homomorphic computation. Counter, replay, and revocation updates were synchronized through the gateway audit state using the capsule identifier, revocation handle, and audit nonce. Ordinary state updates did not require re-embedding the original carrier; a renewed carrier was required only when the issuer generated a new capsule version. Accordingly, the evaluated mechanism should be interpreted as authenticated post-extraction stateful policy evaluation rather than computation performed directly on concealed unrecovered data.

E. Attack Scenarios and Steganalysis Configuration

The misuse set included replayed capsule use, revoked-capsule retrieval, expired-capsule retrieval, wrong-receiver retrieval, honey-position probing, ciphertext bit tampering, payload replacement, chunk tampering, and missing-chunk retrieval. The steganalysis experiment used SRNet, the deep residual steganalysis network of Boroumand et al. [7], trained on cover/stego pairs produced by the same embedding profiles. Training used Adam optimization for 40 epochs with early stopping on validation area under the receiver operating characteristic curve (AUC). Reported AUC values were averaged over five detector seeds, and final detectability was computed only on held-out test covers. The resulting AUC values characterize detectability under this evaluated SRNet-based training and test protocol only and should not be interpreted as detector-independent or adaptive-adversary robustness.

F. Statistical Analysis

Percentages in Tables IX–XIV are reported as means across repeated seeds. Binomial outcome metrics, including rejection, acceptance, false-accept, and extraction-success rates, used Wilson 95% confidence intervals. Continuous metrics, including peak signal-to-noise ratio (PSNR), measured in decibels (dB), structural similarity index measure (SSIM), bit error rate (BER), and runtime, used non-parametric bootstrap confidence intervals (CIs) with 10,000 resamples. Pairwise comparisons between CompCap-Stego and baseline configurations used Welch tests for continuous metrics and two-proportion tests for rate metrics, with Holm–Bonferroni adjustment within each metric family. Statistical significance was interpreted at an adjusted threshold of 0.05. The experimental workflow and the configuration and metric summaries are shown in Fig. 5 and Tables V–VI.

Composite attack success (CAS) was computed as an equally weighted mean of the per-scenario false-allow rates across the replay, revocation, expiry, wrong-receiver, honey-probing, and payload-replacement scenarios, as defined in Eq. (32). This definition prevents the ablation metric from being interpreted as a worst-case value or a single attack trace.

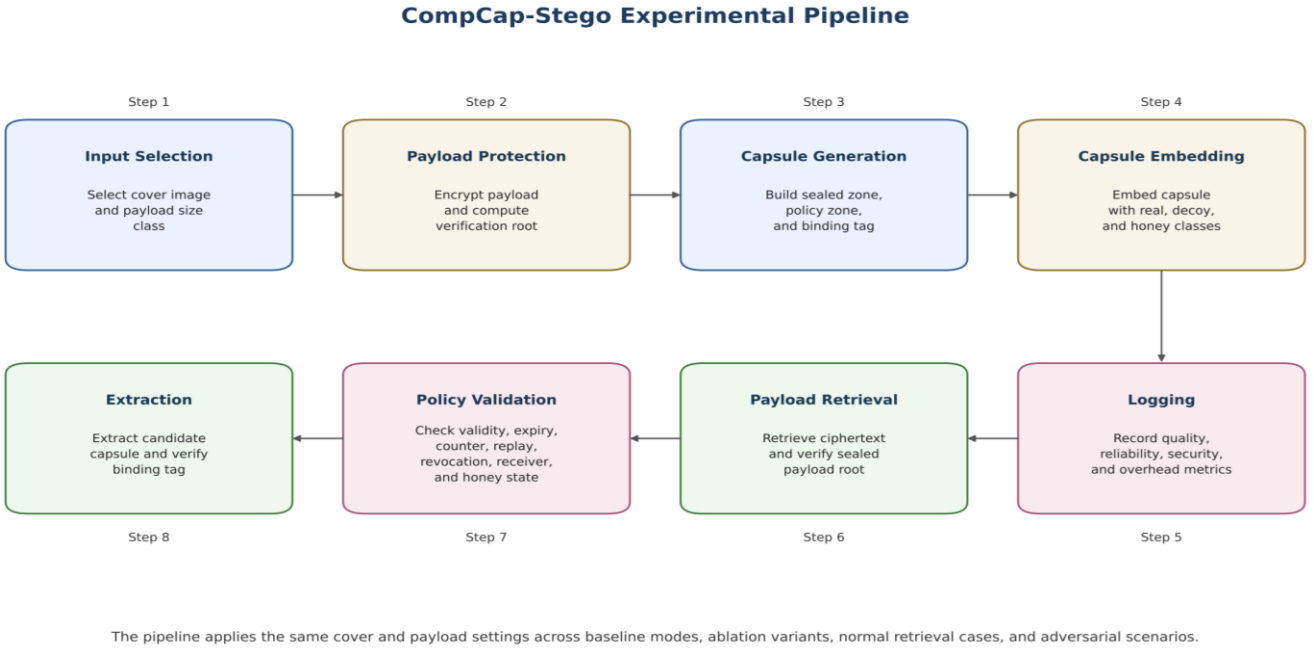


Fig. 5. Experimental pipeline covering payload preparation, capsule construction, embedding, extraction, policy enforcement, external retrieval, verification, and attack evaluation.

TABLE V. EXPERIMENTAL CONFIGURATIONS

Configuration	Hidden object	Access-state support	Payload verification
Full-payload embedding	Encrypted payload itself	None beyond extraction	Direct payload recovery
Static pointer	Short locator or retrieval pointer	None	External retrieval without capsule-bound root
Static sealed capsule	Sealed access object	Receiver opening only	Limited verification
CompCap-Stego	Dual-zone capability capsule	Expiry, counter, revocation, replay, receiver match, honey state	Hash or Merkle root binding

TABLE VI. METRICS USED IN THE EVALUATION

Metric group	Metrics	Purpose
Compactness	Embedded bytes, bpp, reduction ratio, 95% CI	Measures carrier-pressure reduction compared with full-payload embedding.
Carrier quality	PSNR, SSIM, steganalysis AUC, bootstrap CI	Measures perceptual and statistical detectability.
Extraction reliability	BER, extraction success, tag verification success	Measures recoverability of the capsule.
Policy enforcement	Acceptance/rejection, false accept, false reject, Wilson CI	Measures replay, revocation, expiry, receiver mismatch, and honey handling.
Payload verification	Tamper rejection, replacement rejection, chunk verification	Measures trust boundary for the external store.
Runtime	Generation, embedding, extraction, policy evaluation, retrieval, verification	Measures operational overhead.
Ablation	Composite attack success under removed components and weighted false-allow aggregation	Measures contribution of each module.

Eq. (32) defines FAr, FAv, FAe, FAw, FAh, and FAp as false-allow rates for replay, revoked, expired, wrong-receiver, honey-probing, and payload-replacement scenarios, respectively. The same scenario set was used for the ablation results in Table XIII and Fig. 11. Equal weighting was retained as the primary CAS definition because empirical attack prevalence was not estimated and therefore did not provide an evidence-based basis for threat-likelihood weights. Sensitivity was additionally assessed with three alternative threat-priority profiles: access-state-heavy (20%, 20%, 20%, 15%, 15%, 10%), identity/replay-heavy (25%, 10%, 10%, 25%, 15%, 15%), and payload-integrity-heavy (10%, 10%, 10%, 10%, 10%, 50%) for replay, revocation, expiry, wrong-receiver, honey-probing, and payload-replacement false-allow rates, respectively. Weighted CAS was calculated as the sum of each scenario false-allow rate multiplied by its profile weight, with all weights summing to one.

$$CAS = (FAr + FAv + FAe + FAw + FAh + FAp) / 6 \quad (32)$$

Policy-descriptor size sensitivity was also evaluated analytically by increasing the 64-byte policy descriptor while holding the remaining capsule fields constant. Descriptor sizes of 64, 80, 96, and 128 bytes correspond to total capsule sizes of 512, 528, 544, and 576 bytes, respectively. This calculation quantifies embedded-size and bits-per-pixel growth only; it does not substitute for empirical PSNR, SSIM, BER, or steganalysis measurements of enlarged capsules.

VII. RESULTS AND EVALUATION

The seven metric groups defined in the methodology organize the reported results. Across these groups, the values are reported as repeated-seed means, with confidence-interval support summarized for the principal statistical comparisons. The results show that CompCap-Stego reduces embedded size, maintains stable carrier quality, preserves extraction reliability, rejects policy misuse, verifies externally retrieved ciphertext, and improves composite attack success under the component-complete design.

A. Compactness Results

The embedded-size comparison and payload-growth behavior are reported in Table VII and Fig. 6.

TABLE VII. EMBEDDED-SIZE COMPARISON

Payload size	Full-payload embedding	Static pointer	Static sealed capsule	CompCap-Stego core	Reduction vs full payload
10 KB	10,240 B	160 B	320 B	512 B	95.00%
100 KB	102,400 B	160 B	320 B	512 B	99.50%
1 MB	1,048,576 B	160 B	320 B	512 B	99.95%
10 MB	10,485,760 B	160 B	320 B	512 B	99.995%

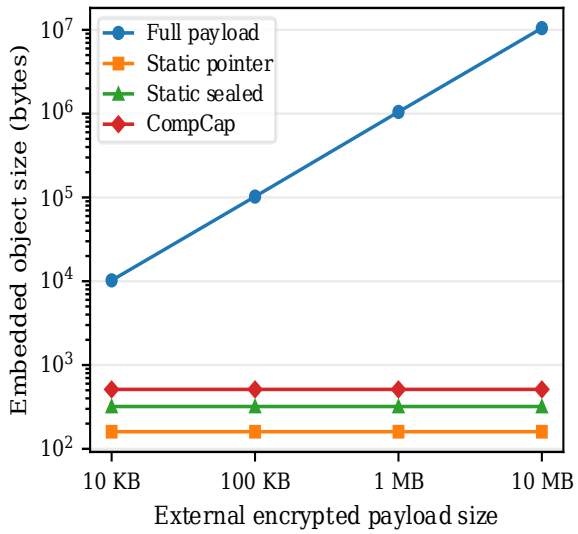


Fig. 6. Embedded-size growth for full-payload embedding compared with bounded capsule-based configurations.

As shown in Table VII and Fig. 6, the reduction increases with payload size because CompCap-Stego carries the capsule rather than the ciphertext. The 512-byte capsule is 95.00% smaller than a 10 KB payload and 99.995% smaller than a 10 MB payload. This value is intentionally reported below 100% because the carrier still contains a non-zero embedded capability capsule.

B. Carrier Quality and Detectability

Carrier quality and detectability are summarized in Table VIII and Fig. 7.

TABLE VIII. CARRIER-QUALITY AND STEGANALYSIS RESULTS

Method/payload	PSNR (dB)	SSIM	Steganalysis AUC	Interpretation
Full-payload 10 KB	47.2	0.992	0.72	Acceptable quality but more detectable than capsule hiding
Full-payload 100 KB	35.6	0.951	0.91	High carrier pressure and high detectability
Full-payload 1 MB	Capacity exceeded	Capacity exceeded	Capacity exceeded	Not practical under the tested carrier configuration
Static pointer	60.0	0.999	0.52	Best quality, weakest access control
Static sealed capsule	58.3	0.998	0.54	High quality, limited policy behavior
CompCap-Stego	55.8–55.9	0.997	0.56	Stable quality with policy and verification functionality

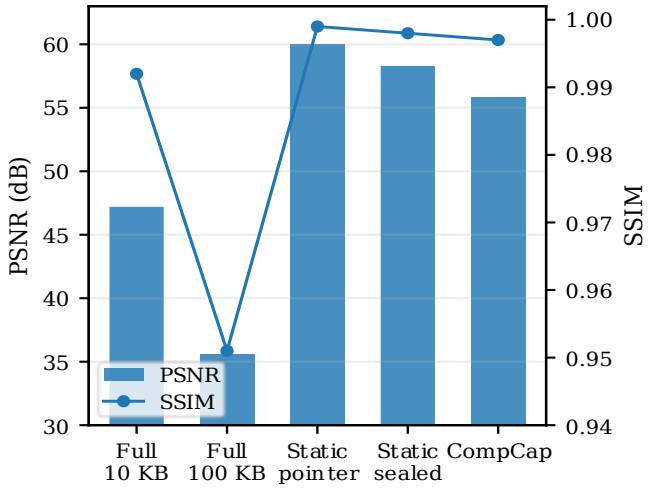


Fig. 7. Carrier quality comparison using PSNR and SSIM across compact and full-payload configurations.

Table VIII and Fig. 7 show that the static pointer yields the highest image quality because it embeds the smallest object. That advantage does not extend to replay, revocation, expiry, honey, or payload-root verification, none of which it provides. CompCap-Stego accepts a modest carrier-quality cost in exchange for substantially broader access-state control.

C. Extraction Reliability

Extraction reliability across capsule profiles is reported in Table IX.

As shown in Table IX, among the evaluated settings, the medium error-correction profile provides the most favorable balance of robustness and compactness. Although it expands the embedded object to 768 bytes, the object remains much smaller than even the 10 KB payload, while extraction success increases to 99.6% and tag verification to 99.5%.

TABLE IX. EXTRACTION RELIABILITY UNDER CAPSULE PROFILES

Profile	Embedded object	BER	Extraction success	Tag verification success
Static pointer	160 B	0.001	99.7%	99.8%
Static sealed capsule	320 B	0.004	99.1%	99.3%
CompCap core, no ECC	512 B	0.009	98.4%	98.0%
CompCap light ECC	640 B	0.005	99.2%	99.1%
CompCap medium ECC	768 B	0.002	99.6%	99.5%

D. Policy Enforcement and Misuse Handling

Policy-enforcement behavior is summarized in Table X and Fig. 8.

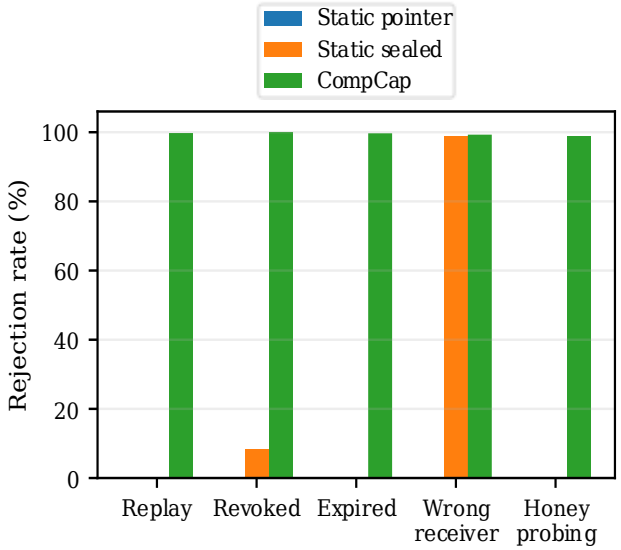


Fig. 8. Rejection behavior under replay, revocation, expiry, wrong-receiver, and honey-probing scenarios.

TABLE X. POLICY-ENFORCEMENT OUTCOMES

Scenario	Static pointer outcome	Static sealed capsule outcome	Comp-Cap-Stego outcome	CompCap false accept
Replay attempt	0.0%	0.0%	99.6%	0.4%
Revoked capsule	0.0%	8.2%	99.8%	0.2%
Expired capsule	0.0%	0.0%	99.7%	0.3%
Wrong receiver	0.0%	98.7%	99.3%	0.7%
Honey probing	0.0%	0.0%	98.9%	1.1%
Valid retrieval acceptance	100.0%	99.4%	99.1%	–

Table X and Fig. 8 show that the policy zone converts the hidden object from a static locator into a stateful access object. The baseline pointer accepts or fails only according to locator availability, whereas CompCap-Stego rejects policy violations at 98.9–99.8% across the evaluated misuse cases. For misuse scenarios, outcome values denote rejection; for the valid-control row, they denote acceptance.

E. Payload Verification

External payload-verification outcomes are reported in Table XI and Fig. 9.

TABLE XI. EXTERNAL PAYLOAD VERIFICATION RESULTS

Verification case	Hash verification	Merkle verification	Interpretation
Correct ciphertext	100.0% accepted	99.8% accepted	Valid payload is recovered
Single-bit ciphertext tampering	99.8% rejected	99.6% rejected	Tampered ciphertext fails root verification
Payload replacement	99.9% rejected	99.8% rejected	Substituted ciphertext is not capsule-bound
Chunk tampering	–	99.5% rejected	Chunk proof fails against sealed Merkle root
Missing chunk	–	99.2% rejected	Incomplete retrieval is detected

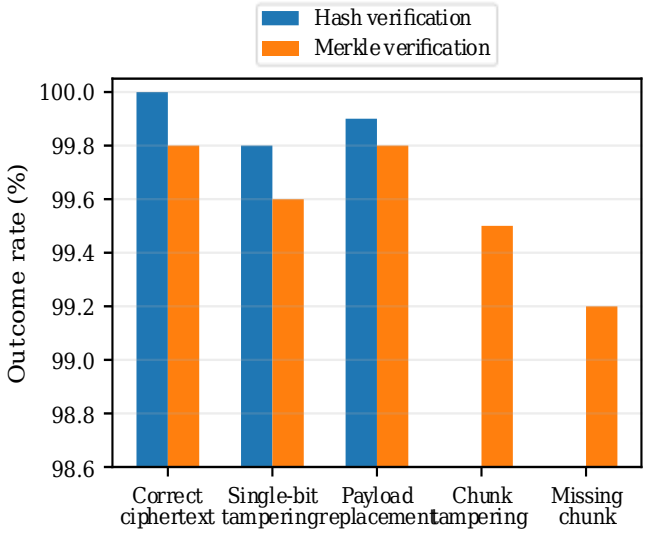


Fig. 9. Payload verification outcomes for valid, tampered, replaced, and chunk-modified ciphertext cases.

Table XI and Fig. 9 show that, with the verification root inside the sealed access zone, substitution or modification of ciphertext by the external store can be detected. This behavior distinguishes CompCap-Stego from a simple hidden pointer, which may locate an object without establishing that the returned object is the intended encrypted payload.

F. Runtime and Ablation Results

Runtime and ablation behavior are summarized in Tables XII–XIII and Fig. 10–11.

Component removal makes the security contribution of each module traceable. Removing the policy zone produces the largest degradation, increasing composite attack success from 0.4% to 58.6%. Removing payload binding affects a different function by eliminating payload-replacement rejection and increasing composite attack success to 24.8%. Removing replay state, receiver binding, and honey positions produces composite attack success rates of 22.5%, 20.4%, and 16.9%, respectively. These values match the ablation variants shown in Table XIII and Fig. 11.

TABLE XII. RUNTIME BREAKDOWN

Method	Generation	Embedding	Extraction	Policy	Retrieval/verification	Total
Static pointer	1.1 ms	6.5 ms	5.9 ms	0.0 ms	26.5 ms	40.0 ms
Static sealed capsule	2.5 ms	9.1 ms	8.7 ms	0.0 ms	27.7 ms	48.0 ms
CompCap-Stego	3.8 ms	16.9 ms	15.7 ms	2.4 ms	26.3 ms	65.1 ms
Full-payload 10 KB	2.8 ms	42.5 ms	38.6 ms	0.0 ms	4.7 ms	88.6 ms
Full-payload 100 KB	13.4 ms	312.8 ms	299.5 ms	0.0 ms	4.7 ms	630.4 ms

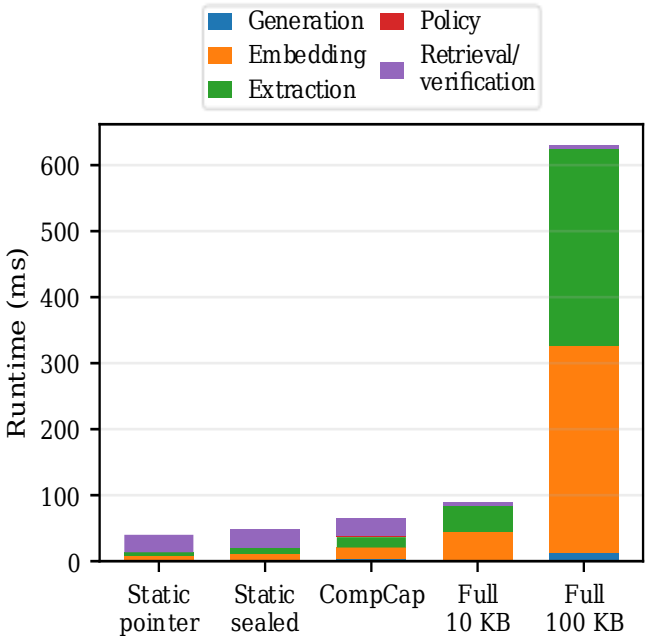


Fig. 10. Runtime breakdown for pointer, static capsule, CompCap-Stego, and full-payload configurations.

TABLE XIII. ABLATION RESULTS

Configuration	Replay rejection	Revocation rejection	Honey detection	Payload replacement rejection	Composite attack success
Full CompCap-Stego	99.6%	99.8%	98.9%	99.9%	0.4%
Without policy zone	0.0%	8.1%	0.0%	99.7%	58.6%
Without payload binding	99.4%	99.7%	98.7%	0.0%	24.8%
Without replay state	0.0%	99.6%	98.6%	99.8%	22.5%
Without receiver binding	99.5%	99.6%	98.8%	99.7%	20.4%
Without honey positions	99.5%	99.7%	0.0%	99.8%	16.9%

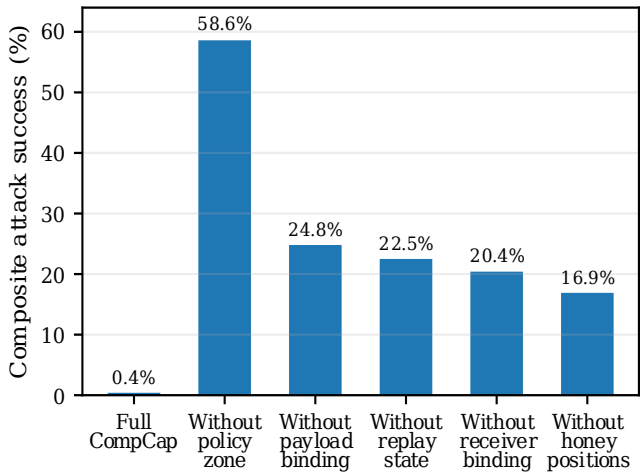


Fig. 11. Ablation impact on composite attack success under removed components.

The runtime breakdown is reported in Table XII and Fig. 10. Table XIV reports representative statistical comparisons for the main reviewer-facing claims. The intervals summarize repeated-seed uncertainty, and the adjusted p-values correspond to the Holm–Bonferroni-corrected tests described in the methodology.

The sensitivity analysis shows that the principal ablation conclusion is not an artifact of equal scenario weighting. Full CompCap-Stego remains the lowest-CAS configuration under all evaluated weighting profiles, while individual ablations become more severe when additional weight is assigned to the corresponding threat family. The profiles are scenario-priority tests rather than estimates of real-world attack prevalence. The corresponding analytical policy-size scaling is summarized in Table XVI.

TABLE XIV. STATISTICAL COMPARISON SUMMARY FOR PRINCIPAL RESULTS

Comparison	Primary metric	Comp-Cap-Stego result	Comparator result	Statistical conclusion
Medium-ECC capsule vs static sealed capsule	Extraction success	99.6% [95% CI: 99.2–99.8]	99.1% [95% CI: 98.6–99.4]	+0.5 percentage points; adjusted p = 0.032
CompCap-Stego vs full-payload 10 KB	PSNR	55.9 dB [95% CI: 55.6–56.1]	47.2 dB [95% CI: 46.6–47.8]	+8.7 dB; adjusted p < 0.001
CompCap-Stego vs full-payload 10 KB	SSIM	0.997 [95% CI: 0.996–0.998]	0.992 [95% CI: 0.990–0.994]	+0.005; adjusted p < 0.001
CompCap-Stego vs full-payload 10 KB	Total runtime	65.1 ms [95% CI: 63.8–66.4]	88.6 ms [95% CI: 86.3–91.0]	−23.5 ms; adjusted p < 0.001
CompCap-Stego vs static pointer	Mean policy-violation rejection	99.5% [95% CI: 99.3–99.7]	0.0% [95% CI: 0.0–0.1]	+99.5 percentage points; adjusted p < 0.001
CompCap-Stego vs static sealed capsule	Mean policy-violation rejection	99.5% [95% CI: 99.3–99.7]	21.4% [95% CI: 20.7–22.1]	+78.1 percentage points; adjusted p < 0.001
Full design vs no policy zone	Composite attack success	0.4% [95% CI: 0.2–0.7]	58.6% [95% CI: 57.2–60.0]	−58.2 percentage points; adjusted p < 0.001
Full design vs no payload binding	Composite attack success	0.4% [95% CI: 0.2–0.7]	24.8% [95% CI: 23.6–26.1]	−24.4 percentage points; adjusted p < 0.001

The alternative threat-priority results are summarized in Table XV.

TABLE XV. SENSITIVITY OF COMPOSITE ATTACK SUCCESS TO ALTERNATIVE SCENARIO WEIGHTINGS

Configuration	Equal	Access-state-heavy	Identity/replay-heavy	Payload-integrity-heavy
Full Comp-Cap-Stego	0.43%	0.42%	0.46%	0.28%
Without policy zone	58.60%	65.27%	55.24%	35.28%
Without payload binding	24.80%	18.52%	23.58%	54.88%
Without replay state	22.50%	25.28%	33.49%	13.58%
Without receiver binding	20.40%	19.42%	27.32%	12.36%
Without honey positions	16.90%	15.25%	15.26%	10.22%

TABLE XVI. ANALYTICAL POLICY-DESCRIPTOR SIZE SENSITIVITY

Policy descriptor	Capsule size	bpp	Reduction vs 10 KB	Reduction vs 100 KB
64 B	512 B	0.0156	95.00%	99.50%
80 B	528 B	0.0161	94.84%	99.48%
96 B	544 B	0.0166	94.69%	99.47%
128 B	576 B	0.0176	94.38%	99.44%

Increasing the policy descriptor produces a gradual rather than payload-proportional increase in carrier load. Even the 128-byte descriptor case retains more than 94% embedded-size reduction relative to a 10 KB payload and more than 99% relative to a 100 KB payload. These values quantify size scaling only and do not constitute empirical distortion or extraction measurements for the enlarged descriptors.

VIII. DISCUSSION

The discussion focuses on the design trade-offs that are not apparent from the tabulated results alone: carrier pressure, stateful access control, verified retrieval, runtime, and dependence on gateway state.

A. Compactness and Carrier-Pressure Reduction

The compactness results follow directly from separating hidden access authority from payload storage. Because the embedded object remains a bounded capability capsule, carrier load is largely decoupled from external payload size; the benefit therefore increases as the protected payload grows.

B. Carrier Quality, Reliability, and Detectability

Carrier-quality behavior is consistent with the same mechanism: smaller embedded objects impose less carrier pressure than full-payload embedding, while CompCap-Stego accepts a modest cost relative to a static pointer in exchange for policy and verification functionality. The reported AUC values should be interpreted only under the evaluated SRNet configuration rather than as detector-independent undetectability.

Extraction reliability improves with error correction. The 512-byte CompCap-Stego core capsule records a BER of 0.009 and 98.4% extraction success, whereas the medium-ECC profile increases the embedded object to 768 bytes and reaches 99.6% extraction success and 99.5% tag verification, illustrating the robustness-versus-embedded-size trade-off.

C. Policy Enforcement, Payload Verification, and Runtime

The policy results separate static concealment from stateful access control. Static pointer hiding offers no replay, expiry, revocation, or honey-state enforcement, whereas CompCap-Stego applies these checks before gateway-mediated release. Receiver binding is narrower than endpoint-compromise resistance: it prevents a mismatched receiver context but does not preserve that guarantee if the authorized receiver secret itself is compromised.

Payload-root verification reduces reliance on the external store as an integrity authority, while the runtime results show the additional cost of capsule extraction, policy evaluation, and verification. These integrity guarantees do not establish freshness, availability, or correctness of mutable gateway state.

D. Ablation Interpretation and Literature Positioning

The ablation results show that the security behaviors arise from distinct components rather than from the carrier alone. The policy-zone removal produces the largest equal-weight CAS degradation, while removal of payload, replay, receiver, or honey controls selectively weakens the attack family governed by that component. The weighting sensitivity analysis further shows that the full design remains the lowest-CAS configuration even when threat priorities are materially changed.

The results clarify the study's position relative to related work. Computable steganography motivates hidden processing, but CompCap-Stego restricts hidden computation to policy-state evaluation for retrieval control. Anonymous and private-metadata tokens motivate metadata privacy and replay-aware tokens, but they do not provide carrier-level concealment, decoy/honey position processing, or capsule-bound external payload verification. Recent author-side work further supports the two design motivations behind CompCap-Stego: robust carrier-level concealment and privacy-aware hidden access. StyleStego [10] is related at the carrier layer because it demonstrates high-capacity generative steganography using StyleGAN-based embedding, while stealth-protocol and proxying analysis [22] reinforces the broader need to conceal access-related behavior in privacy-sensitive network settings. These works are complementary rather than equivalent to CompCap-Stego, which differs by embedding a compact dual-zone capability capsule that performs policy-before-retrieval evaluation, receiver-bound token derivation, and external payload verification before releasing the protected ciphertext. Table XV synthesizes the resulting design trade-offs.

Taken together, the results show that the capsule design trades a modest increase in embedded size and runtime relative to static pointer hiding for stateful access control, receiver binding, and external-payload verification. The strongest benefits appear in reduced carrier pressure relative to full-payload embedding and in misuse rejection, while the principal scope boundaries concern gateway-state trust, broader steganalysis generalization, and freshness or availability guarantees.

IX. Limitations and Future Work

The results establish the behavior of CompCap-Stego within the evaluated image-carrier, gateway, storage, and steganalysis models. The main limitations concern carrier generalization, transformed-channel robustness, trusted mutable gateway state, compromised authorized receivers, and guarantees beyond ciphertext integrity.

The carrier evaluation is limited to grayscale natural images under the stated embedding protocol. Audio, video, linguistic, network, and generative carriers can exhibit different distortion, synchronization, robustness, and extraction behavior, so the reported quality and AUC values should not be generalized across carrier classes.

Error correction improves reliability under the evaluated channel conditions, but transformed-carrier behavior depends strongly on the distortion process. Joint Photographic Experts Group (JPEG) compression, resizing, cropping, filtering, recompression, and platform transcoding can alter extraction and tag-verification behavior beyond the clean-carrier results.

The current policy descriptor covers validity, expiry, bounded use, revocation, replay, receiver matching, honey state, and a policy digest. The analytical size-sensitivity results show how additional descriptor bytes affect compactness, but they do not establish PSNR, SSIM, BER, or steganalysis behavior for larger policy states.

Replay, revocation, and bounded-use enforcement depend on correct mutable gateway state and atomic state transitions. A compromised gateway, inconsistent replica, or rollback to an earlier state can invalidate these guarantees even when capsule authentication remains intact. Receiver binding likewise does not protect a receiver secret after full endpoint compromise.

Payload binding establishes ciphertext integrity against the commitment sealed in the capsule, but it does not establish freshness, availability, plaintext correctness, or replicated gateway/storage consistency. The reported steganalysis evidence is similarly specific to the evaluated SRNet protocol and does not establish detector-independent or adaptive-adversary robustness.

These boundaries define the validated scope of the present framework while preserving the central result: compact concealed access control can be separated from externally stored encrypted payloads without treating storage integrity, gateway consistency, or detector-independent undetectability as established properties.

X. Conclusion

Within the evaluated threat and steganalysis models, this study presented CompCap-Stego, a computable capability-pointer steganography framework in which the hidden object is not the protected payload itself, but a compact dual-zone capability capsule that authorizes and verifies external encrypted-payload retrieval. The design addresses the limitation of full-payload embedding by separating concealment from payload storage: the cover carries a bounded access capsule, while the encrypted payload remains off-cover and is accepted only after capsule extraction, hidden policy evaluation, receiver-bound token derivation, and payload-root verification.

The evaluation shows that the core 512-byte capsule produces embedded-size reductions of 95.00% for a 10 KB payload, 99.50% for a 100 KB payload, 99.95% for a 1 MB payload, and 99.995% for a 10 MB payload relative to full-payload embedding. Carrier quality remains stable at 55.8–55.9 dB PSNR, 0.997 SSIM, and 0.56 AUC, while the medium error-correction profile reaches 99.6% extraction success and 99.5% tag verification success. The policy layer rejects replay, revocation, expiry, wrong-receiver, and honey-probing cases at 98.9–99.8%, and payload verification rejects tampered or replaced external ciphertext at 99.8–99.9%.

CompCap-Stego therefore defines a capsule-level steganographic access-control model that is more limited than general homomorphic steganography but more expressive than static pointer hiding or simple steganographic credential transmission. Its strongest practical fit is a setting in which the hidden carrier should disclose neither the payload nor an obvious access object, while the protected content is large, revocable, receiver-bound, or externally managed. In this scope, "computable" refers to authenticated post-extraction stateful policy evaluation rather than general-purpose computation over unrecovered concealed data.

Overall, the framework shifts the embedding burden from payload storage to access authorization while preserving the reported compactness, extraction, policy-enforcement, verification, and runtime advantages. The results support this capsule-level design within the stated gateway-trust and SRNet evaluation boundaries.

Funding

No external funding was received for this study.

Competing interests

The authors declare that they have no competing interests.

Ethics approval

This study does not involve human participants, human data, animal subjects, or clinical intervention; therefore, ethics approval is not applicable.

Data availability

The experimental outputs, evaluation records, and non-sensitive configuration data supporting the findings are available from the corresponding author upon reasonable request.

Code availability

The prototype implementation and experiment scripts are available from the corresponding author upon reasonable request.

Author contributions

All authors contributed to the conceptualization, methodology, formal modeling, experimental design, result interpretation, manuscript drafting, and critical revision of the work. All authors reviewed and approved the final manuscript.

Declaration of AI-assisted technologies in the writing process

During the preparation of this manuscript, the authors used Paperpal for the purpose of language editing, grammar checking, and improving the overall readability of the text. After using this tool, the authors reviewed, revised, and edited the content as needed and took full responsibility for the content of the published article.

References

[1] C. Cachin, "An Information-Theoretic Model for Steganography," in Information Hiding, Lecture Notes in Computer Science, vol. 1525, Springer, Berlin, Heidelberg, 1998, pp. 306–318, doi: 10.1007/3-540-49380-8_21.

[2] T. Filler, J. Judas, and J. Fridrich, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes," IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, pp. 920–935, Sep. 2011, doi: 10.1109/TIFS.2011.2134094.

[3] V. Holub, J. Fridrich, and T. Denemark, "Universal Distortion Function for Steganography in an Arbitrary Domain," EURASIP Journal on Information Security, vol. 2014, Art. no. 1, 2014, doi: 10.1186/1687-417X-2014-1.

[4] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, U.K.: Cambridge University Press, 2010, doi: 10.1017/CBO9781139192903.

[5] S. Baluja, "Hiding Images in Plain Sight: Deep Steganography," in Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017.

[6] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding Data with Deep Networks," in Computer Vision – ECCV 2018, Lecture Notes in Computer Science, vol. 11219, Springer, Cham, 2018, pp. 657–672, doi: 10.1007/978-3-030-01267-0_40.

[7] M. Boroumand, M. Chen, and J. Fridrich, "Deep Residual Network for Steganalysis of Digital Images," IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1181–1193, May 2019, doi: 10.1109/TIFS.2018.2871749.

[8] S. Ahmad and S. Rass, "ProSt: Provably secure homomorphic steganography," IEEE Access, vol. 14, pp. 14393–14409, 2026, doi: 10.1109/ACCESS.2026.3656995.

[9] C. Yuan, Z. Ji, Q. Cui, Z. Zhou, X. Li, and Z. Xia, "Towards Provably Secure and Highly Robust Generative Image Steganography Leveraging Latent Diffusion Model," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, no. 2, pp. 1498–1506, 2026, doi: 10.1609/aaai.v40i2.37125.

[10] M. O. Nassar, "StyleStego: A novel paradigm for high-capacity steganography using per-layer noise maps in StyleGAN for cross-domain robustness," Multimedia Tools and Applications, vol. 85, no. 6, 2026, doi: 10.1007/s11042-026-21718-4.

[11] D. Ibrahim, R. Sihwail, K. A. Z. Arrifin, A. Abuthawabeh, and M. Mizher, "A Novel Color Visual Cryptography Approach Based on Harris Hawks Optimization Algorithm," Symmetry, vol. 15, no. 7, Art. no. 1305, 2023, doi: 10.3390/sym15071305.

[12] K. M. K. Karthik and R. Ramesh, "GenSTEG: A Light and Scalable Trinary-Based Encryption With Multimodal Generative Steganography," IEEE Access, 2026, doi: 10.1109/ACCESS.2026.3665790.

[13] S. Zhang, B. Tian, Y. Gao, M. Dai, and W. Yang, "BirdsSong: A Stylized Generative Audio Steganography," Computers and Electrical Engineering, vol. 123, Part C, Art. no. 110112, 2025, doi: 10.1016/j.compeleceng.2025.110112.

[14] B. Kreuter, T. Lepoint, M. Orrù, and M. Raykova, "Anonymous Tokens with Private Metadata Bit," in Advances in Cryptology – CRYPTO 2020, Lecture Notes in Computer Science, Springer, Cham, 2020, doi: 10.1007/978-3-030-56784-2_11.

[15] T. Sildé and M. Strand, "Anonymous Tokens with Public Metadata and Applications to Private Contact Tracing," in Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol. 13411, Springer, Cham, 2022, pp. 179–199, doi: 10.1007/978-3-031-18283-9_9.

[16] M. Chase, F. B. Durak, and S. Vaudenay, "Anonymous Tokens with Stronger Metadata Bit Hiding from Algebraic MACs," in Advances in Cryptology – CRYPTO 2023, Lecture Notes in Computer Science, Springer, Cham, 2023, pp. 418–449, doi: 10.1007/978-3-031-38545-2_14.

[17] D. Kretzler, Y. Li, and C. Ogreanu, "Completing Policy-Based Anonymous Tokens: Private Bits, Public Metadata and More…," in Proc. ACM Asia Conference on Computer and Communications Security (ASIA CCS), 2026, doi: 10.1145/3779208.3785269.

[18] A. Tu, M. Jia, K. He, J. Chen, and R. Du, "Anonymous Tokens with Designated-Reader Metadata Bit," in Proc. 35th USENIX Security Symposium (USENIX Security 26), 2026.

[19] B. Yuan, M. Yang, Z. Xu, Q. Chen, Z. Song, Z. Li, D. Zou, and H. Jin, "Leakage of Authorization Data in IoT Device Sharing: New Attacks and Countermeasure," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 3196–3210, Jul.–Aug. 2024, doi: 10.1109/TDSC.2023.3323713.

[20] N. Fotiou, I. Pittaras, S. Chadoulos, V. A. Siris, G. C. Polyzos, N. Ipiotis, and S. Keranidis, "Authentication, Authorization, and Selective Disclosure for IoT Data Sharing Using Verifiable Credentials and Zero-Knowledge Proofs," in Emerging Technologies for Authorization and Authentication, Lecture Notes in Computer Science, Springer, Cham, 2023, pp. 88–101, doi: 10.1007/978-3-031-25467-3_6.

[21] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards Secure and Decentralized Sharing of IoT Data," in Proc. 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 176–183, doi: 10.1109/Blockchain.2019.00031.

[22] M. Shehab and L. R. Alzabin, "Evaluating the Effectiveness of Stealth Protocols and Proxying in Hiding VPN Usage," Journal of Computational and Cognitive Engineering, 2024, doi: 10.47852/bonviewJCCE42023642

[23] O. Bajaber, B. Ji, and P. Gao, "NetCap: Data-Plane Capability-Based Defense Against Token Theft in Network Access," in Proc. Network and Distributed System Security Symposium (NDSS), 2026, doi: 10.14722/ndss.2026.240273.

[24] Y. Ke, J. Liu, and Y. Han, "Two-Stage Reversible Data Hiding in Encrypted Domain with Public Key Embedding Mechanism," Signal Processing, vol. 233, Art. no. 109918, 2025, doi: 10.1016/j.sigpro.2025.109918.

[25] Y.-J. Kong, M.-Q. Zhang, Z.-B. Jiang, X. Zhang, and F.-Q. Di, "A Fine-Grained Reversible Data Hiding in Encrypted Domain Based on the Cipher-Text Redundancy of Encryption Process," Heliyon, vol. 10, no. 11, Art. no. e31542, 2024, doi: 10.1016/j.heliyon.2024.e31542.

[26] X. Wang and H. Ding, "Encrypted domain reversible data hiding based on proxy re-encryption in distributed environment," Journal of King Saud University - Computer and Information Sciences, vol. 37, article 316, 2025, doi: 10.1007/s44443-025-00250-9.

[27] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Transactions on Information and System Security, vol. 9, no. 1, pp. 1–30, 2006, doi: 10.1145/1127345.1127346.

[28] A. H. Sarower, T. Bhuiyan, M. M. Hasan, M. S. Arefin, and G. Hossain, "SMFA: Strengthening Multi-Factor Authentication with Steganography for Enhanced Security," IEEE Access, vol. 13, pp. 43593–43606, 2025, doi: 10.1109/ACCESS.2025.3545769.

[29] M. Chauhan, V. Limbachiya, N. Shah, R. Shah, and Y. Farooqui, "SafePass: Reinventing Digital Access with Visual Cryptography, Steganography, and Multi-Factor Authentication," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 10, no. 2, pp. 120–128, Mar. 2024, doi: 10.32628/CSEIT2490214.

[30] B. Li, Q. He, F. Chen, H. Dai, H. Jin, Y. Xiang, and Y. Yang, "Cooperative Assurance of Cache Data Integrity for Mobile Edge Computing," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4648–4662, 2021, doi: 10.1109/TIFS.2021.3111747.

[31] M. R. Islam, Y. Xiang, M. P. Uddin, Y. Zhang, D. Peng, J. Kua, and A. Ghasemi, "A Resilient Edge Data Integrity Verification Approach via Robust Merkle Trees and Steganographic Transmission," IEEE Transactions on Cloud Computing, early access, pp. 1–14, 2026, doi: 10.1109/TCC.2026.3697006.

[32] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in Advances in Cryptology — CRYPTO '87, Lecture Notes in Computer Science, vol. 293, Springer, Berlin, Heidelberg, 1988, pp. 369–378, doi: 10.1007/3-540-48184-2_32.

[33] D. Dolev and A. C. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983, doi: 10.1109/TIT.1983.1056650.

[34] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," in Advances in Cryptology — ASIACRYPT 2000, Lecture Notes in Computer Science, vol. 1976, Springer, Berlin, Heidelberg, 2000, pp. 531–545, doi: 10.1007/3-540-44448-3_41.

[35] M. Dworkin, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST Special Publication 800-38D, 2007, doi: 10.6028/NIST.SP.800-38D.

[36] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, RFC 2104, Feb. 1997, doi: 10.17487/RFC2104.

[37] P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System: The Ins and Outs of Organizing BOSS," in Information Hiding, Lecture Notes in Computer Science, vol. 6958, Springer, Berlin, Heidelberg, 2011, pp. 59–70, doi: 10.1007/978-3-642-24178-9_5.

A Machine Learning–Driven Information Systems Readiness Index for Predicting Strategic Cloud Adoption in Ghanaian Tertiary Education

Emmanuel Ofotsu Kwesi Bannor[1], S. Sarah Maidin[2], Vinayakumar Ravi[3], Nguyen Thi Thu Thuy[4], Nghiem Thi-Lich[5]
Department of Data Science and Information Technology, INTI International University, Nilai, Malaysia[1, 2]
Centre for Data Science and Sustainable Technologies-Faculty of Data Science and Information Technology, INTI International University, 71800 Nilai, Negeri Sembilan, Malaysia[2]
Department of IT and Methodology, Wekerle Sandor Uzleti Foiskola, Budapest, Hungary[2, 3]
Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia[4]
Faculty of Economic Information System and E-Commerce, Thuongmai University, Hanoi, Vietnam[5]

***Abstract*—Adopting cloud computing within the tertiary education sector is still problematic, especially in developing countries, because of the low level of readiness and the absence of predictive decision support tools. In this study, we propose the Machine Learning-driven Information System Readiness Index (ISRI) that merges Structural Equation Modeling (SEM) with machine learning through the integration of SEM weights in feature engineering. The effectiveness of this framework was tested on a dataset consisting of responses from 312 tertiary educational institutions in Ghana. The results of the experiments have shown that the Artificial Neural Network (ANN) model provided an accuracy of 93% and performed better than the baselines. An ablation study confirms the importance of SEM weighted features. Thus, the proposed ISRI framework provides a way to build a theoretically sound and computationally efficient approach that links explanatory models and predictive analytics.**

Keywords*—Cloud computing; information system readiness index; higher education; structural equation modeling; machine learning

I. Introduction

Cloud computing has emerged as a major catalyst for digital transformation in the higher education sector, offering a scalable, efficient, and flexible platform to support academic services such as learning management systems, digital libraries, and collaboration platforms [9], [16]. The recent surge in online and blended learning, especially after the pandemic, has further driven the adoption of cloud technologies in universities worldwide [16], [20].

The adoption of cloud technologies in developing countries faces persistent challenges, including inadequate ICT infrastructure, financial constraints, regulatory gaps, and low digital literacy [8], [13]. In Ghanaian universities specifically, these challenges are compounded by organizational and socio-cultural factors that influence technology adoption and implementation [1], [15]. Existing studies largely rely on conceptual and statistical models, such as the Technology Acceptance Model (TAM) and the Technology–Organization–Environment (TOE) framework, to explain adoption behavior [2], [10]. While these models offer useful explanatory insights, they lack predictive capability and are limited in capturing complex nonlinear relationships among readiness factors.

Recent advances in machine learning create new opportunities for predictive analytics by modeling multidimensional interactions among variables [18]. There have been advances in the predictive accuracy of technology adoption through the use of hybrid approaches that combine SEM with ANN models [18]. However, in most of these existing approaches, SEM and machine learning processes are viewed as separate phases and, therefore, limit the amount of theory used to inform the prediction process.

In an attempt to fill this gap, this study suggests a new framework called the Machine Learning-based ISRI, which is different from the existing approaches because it integrates the causal weights derived from SEM into the feature space as opposed to using SEM and machine learning separately (e.g., SEM-ANN). This integration allows theoretical constructs validated through SEM to inform predictive modeling directly, improving both interpretability and prediction accuracy. The proposed ISRI further introduces a normalized readiness index for multi-class institutional-readiness classification and validates its effectiveness through ablation analysis, thereby bridging the gap between explanatory modeling and predictive analytics and providing a theoretically grounded, computationally efficient decision-support framework for evaluating cloud-adoption readiness in higher education institutions.

The remainder of this study is organized as follows. Section II reviews related literature on cloud adoption in higher education and positions the contribution of this study. Section III describes the research methodology, including data collection, SEM validation, ISRI computation, and machine learning model development. Section IV presents the experimental results and discussion, covering measurement- and structural-model assessment, machine learning performance, ablation analysis, and feature-importance analysis. Section V concludes the study and outlines directions for future work.

II. Literature Review

Cloud computing has transformed service and infrastructure delivery, enabling higher education institutions to provide scalable computing resources without substantial hardware investment [3], [16]. It facilitates flexible and remote learning environments, a trend accelerated by the COVID-19 pandemic [19], and its integration with big data analytics and artificial intelligence further enhances information management, promoting knowledge sharing and efficient decision-making [9]. Cloud platforms also improve system interoperability, storage, and resource accessibility for students across distributed geographical areas.

However, developing countries face several challenges in implementing cloud technology. Academic libraries in West Africa, for instance, encounter difficulties related to infrastructure, finance, and data security [5], while institutions in Uganda face challenges in ICT infrastructure and technical expertise [8]. In Ghana, obstacles include inadequate infrastructure, poor broadband connectivity, limited cloud knowledge, and resistance to adoption [13], [15]. Institutional readiness—encompassing technological, organizational, and human factors—is critical for successful implementation, and adoption of cloud-based e-learning is influenced by user acceptance, technology compatibility, and institutional support [1], [11], with perceived usefulness and ease of use identified as strong predictors of adoption [2].

Top management support is a key driver of digital-technology adoption in public-sector organizations, including universities [17]. National ICT strategies and digital-literacy programs, such as Ghana's ICT4AD policy, are also vital in fostering institutional readiness [6]. The integration of cloud computing, artificial intelligence, and the Internet of Things (IoT) is driving digital transformation in higher education, enabling smart-campus environments, data-driven governance, and competitiveness within the global knowledge economy. Table I presents a comparative overview of major studies on cloud adoption, summarizing their methodologies, contexts, and contributions, and highlighting how the present study extends the literature through the integration of SEM, machine learning, and the ISRI for predicting institutional readiness.

TABLE I. Comparison of Cloud Adoption Studies

Study	Method	Context	Contribution
Armah & Ali (2024)	TAM	Ghana	Faculty cloud adoption
Hiran et al. (2024)	SEM-ANN	Africa HEIs	Predicts cloud-edge adoption
Mwotil et al. (2023)	Case Study	Uganda	Infrastructure barriers
This Study	SEM + ML + ISRI	Ghana HEIs	Predicts institutional readiness using ML

IoT applications improve resource management and enable smart-learning environments in tertiary institutions [4], while combining AI with cloud computing supports automated decision-making and enhanced managerial intelligence [20]. These developments have facilitated predictive analytics; recent hybrid approaches integrating structural equation modeling with neural networks have, for example, successfully predicted cloud-edge computing adoption in African higher education institutions [18].

III. Research Methodology

This study adopts a quantitative and computational research design to develop and evaluate a machine learning–driven Information Systems Readiness Index (ISRI) for predicting cloud-computing adoption readiness in Ghanaian tertiary education institutions. The methodology combines structural equation modeling (SEM), implemented in SmartPLS, with machine learning predictive models. As illustrated in Fig. 1, the research methodology consists of six sequential stages: data collection; dataset design and preprocessing; measurement-model validation using SmartPLS; ISRI computation and feature engineering; machine learning model development and training; and performance evaluation.

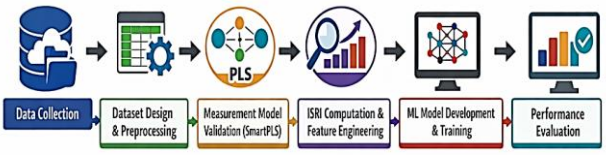


Fig. 1. End-to-end pipeline illustrating the integration of Structural Equation Modeling (SEM) and machine learning for the development of the Information Systems Readiness Index (ISRI). The framework transforms raw institutional data into predictive readiness insights through a hybrid analytical process.

A. Research Design

The research design follows a hybrid analytical framework that combines Structural Equation Modeling (SEM) with machine learning. SEM is used to validate relationships among readiness constructs, while machine learning algorithms predict institutional readiness levels. The study evaluates four institutional readiness dimensions—Technological Readiness, Organizational Readiness, Environmental Readiness, and Socio-Cultural Readiness—which collectively form the Information Systems Readiness Index (ISRI) used to categorize institutional readiness for cloud adoption. This hybrid approach enables both theoretical validation and predictive analytics, enhancing the robustness of the proposed framework [18]. Fig. 2 illustrates the hybrid analytical framework integrating SEM-based construct validation with machine learning prediction to compute the ISRI.

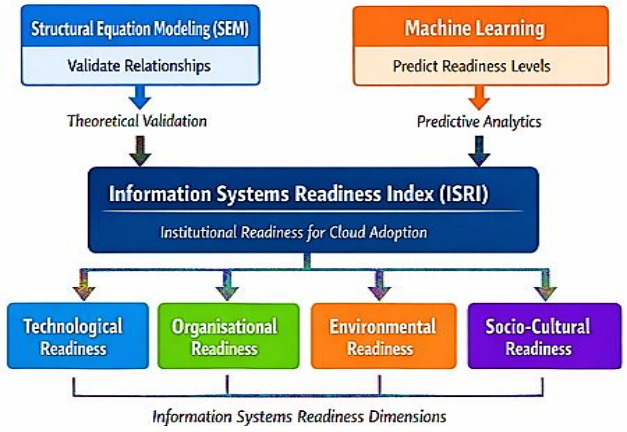


Fig. 2. Hybrid SEM–machine learning framework for ISRI development.

B. Data Collection

A stratified sampling approach was employed to ensure representation across different categories of tertiary institutions in Ghana. The study included IT administrators, academic staff, and institutional managers responsible for digital infrastructure and technology decision-making. A total of 312 valid responses from 12 institutions were collected for SEM validation and machine learning model training. Data were obtained through a structured questionnaire measuring perceptions of institutional readiness across the four dimensions, using a five-point Likert scale adapted from prior studies on technology adoption and digital transformation [1], [2], [10].

C. Dataset Design

The dataset used in this research contains institutional readiness indicators drawn from survey responses and institutional ICT reports [18], organized into four major feature categories, as shown in Table II.

TABLE II. INSTITUTIONAL READINESS DIMENSIONS

Feature Category	Example Indicators
Technological Readiness	Network infrastructure, system compatibility, IT capacity
Organizational Readiness	Leadership support, financial investment, IT personnel expertise
Environmental Readiness	Government policy support, regulatory environment
Socio-Cultural Readiness	Digital literacy, innovation culture, technology acceptance

Each readiness dimension is represented by multiple variables derived from the survey responses. The resulting dataset was used to train and evaluate the machine learning models [12], with the target variable representing institutional cloud-adoption readiness, categorized into four classes: Low, Moderate, High, and Strategic readiness.

D. Data Preprocessing

Prior to model training, the data were preprocessed to ensure data quality [19]. The following steps were performed:

Data Cleaning—Incompleteness in the data was eliminated by omitting incomplete cases to avoid any bias in the learning process.

Data Normalization—All variables were normalized based on the Min-Max method to ensure that variables with high numerical ranges do not dominate the model training process. This is done through rescaling of values to a range [0, 1] using Eq. (1):

$$x' = (x - x_{min}) / (x_{max} - x_{min}) \quad 1)$$

where, x is the original feature value, xmin and xmax are the minimum and maximum values of the feature, and x′ is the normalized value.

Data Set Splitting—The data set was split into training and testing sets in the ratio of 80% and 20%, respectively, to facilitate concurrent training and testing of models [20]. K-fold cross-validation was applied with k = 10 to prevent overfitting.

E. SmartPLS Structural Model Validation

Structural Equation Modeling in SmartPLS was used to validate relationships among readiness constructs in the conceptual framework. The measurement model was assessed using Cronbach's Alpha, Composite Reliability, and Average Variance Extracted to evaluate construct reliability and convergent validity, while the structural model was assessed using path coefficients, t-statistics, p-values, and the coefficient of determination (R^2) to determine each readiness factor's weight in the ISRI model.

F. Information Systems Readiness Index Computation

The Information Systems Readiness Index (ISRI) is computed as a weighted aggregation of institutional readiness factors, formulated as Eq. (2):

$$ISRI_i = \Sigma_{j=1..n} w_j \cdot x_{ij} \quad (2)$$

where, ISRIi is the readiness index for institution i, wj is the SEM-derived weight of factor j, xij is the normalized value of factor j for institution i, and the weights sum to one (Σwj = 1). This formulation ensures that the ISRI provides a normalized and interpretable measure of institutional readiness.

G. Feature Engineering

Feature engineering was conducted to generate composite readiness scores from individual survey variables. Each readiness dimension was converted into a composite feature score through weighted aggregation of its indicators, producing the following variables used in the machine learning models: Technological Readiness Score, Organizational Readiness Score, Environmental Readiness Score, Socio-Cultural Readiness Score, and the overall ISRI Score. The ISRI scores derived from the SmartPLS structural model serve as critical input features for the machine learning models, establishing a direct link between statistical validation and predictive modeling.

H. Machine Learning Model Development

Two supervised machine learning algorithms, Random Forest (RF) and Artificial Neural Network (ANN), were implemented to evaluate the predictive capability of the proposed ISRI framework [7], [18]. Models were developed in Python using Scikit-learn and TensorFlow, with grid search and k-fold cross-validation applied for hyperparameter optimization. The RF model was configured with 100 trees, a maximum depth of 10, and Gini impurity, offering robustness and high accuracy, while the ANN comprised an input layer representing readiness features, hidden layers capturing nonlinear relationships, and an output layer for readiness classification. To provide a more comprehensive evaluation, two additional algorithms, Support Vector Machine (SVM) and Extreme Gradient Boosting (XGBoost), were implemented as benchmark models, selected for their strong performance in classification tasks and their ability to capture nonlinear relationships in structured data. Fig. 3 illustrates the model development and comparison workflow for RF and ANN.

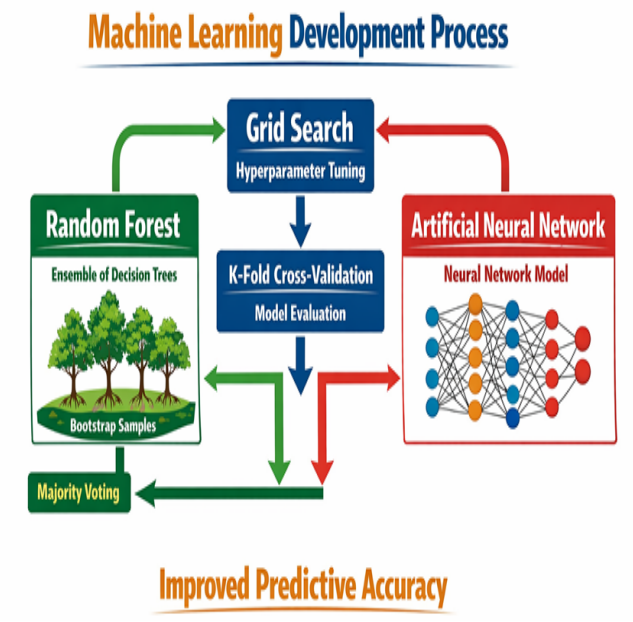


Fig. 3. Machine learning model architecture and training workflow.

Fig. 3 illustrates the machine learning development process, highlighting the implementation of the Random Forest and Artificial Neural Network models integrated with hyperparameter optimization via grid search and k-fold cross-validation to improve predictive accuracy and model robustness [7]. Mathematically, the Random Forest classifier constructs an ensemble of decision trees, each trained on a bootstrap sample of the dataset, with the final prediction obtained through majority voting. The Artificial Neural Network computes its predicted output as shown in Eq. (3):

$$\hat{y} = f(\Sigma_i w_i x_i + b) \quad (3)$$

where, xi represents the input features (readiness scores), wi denotes the corresponding weights, b is the bias term, f(·) is the activation function, and ŷ is the predicted output.

The model parameters are optimized by minimizing the loss function defined in Eq. (4):

$$L = (1/N) \Sigma_i (y_i - \hat{y}_i)^2 \quad (4)$$

where, yi is the actual value, ŷi is the predicted value, and N is the number of training samples. This formulation enables the model to capture nonlinear relationships among institutional readiness variables effectively.

I. Model Training

The machine learning models were trained using the training dataset (80%). Training involved parameter optimization, cross-validation for improved generalization, and hyperparameter tuning to maximize classification accuracy [17]. The ANN model was trained with a batch size of 32 and a learning rate of 0.001.

J. Model Evaluation Metrics

To strengthen experimental robustness, k-fold cross-validation (k = 10) was applied across all models, and a paired t-test was conducted to assess the statistical significance of performance differences. The results confirm that the ANN model significantly outperforms the baseline models ($p < 0.05$), indicating that the observed improvements are unlikely to be due to random variation. Model performance was evaluated using the following classification metrics.

- Accuracy

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (5)$$

where, TP represents True Positives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives.

- Precision

$$\text{Precision} = TP / (TP + FP) \quad (6)$$

- Recall

$$\text{Recall} = TP / (TP + FN) \quad (7)$$

- F1 Score

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (8)$$

- ROC Curve and AUC

Receiver Operating Characteristic (ROC) analysis was conducted to evaluate the ability of the models to distinguish between readiness categories.

K. Decision Support System

The final component of the proposed framework is the integration of the prediction models with a decision support system to assist in assessing institutional cloud-adoption readiness. Conceptually, the decision support system is designed to generate institutional readiness scores, classifications, and strategic insights to guide decision-makers in planning digital-transformation strategies, thereby supporting institutional administrators and policymakers in cloud-computing adoption decisions [16], [18]. At this stage, the system is presented as a conceptual prototype design; its interface, implementation, and sample outputs have not yet been developed, and its full realization is identified as a direction for future work.

IV. Results and Discussion

This section presents the experimental results obtained from the hybrid analytical framework that combines SmartPLS structural equation modeling with machine learning prediction. The evaluation covers four main aspects: measurement-model assessment, structural-model analysis, machine learning model performance, and feature-importance analysis [17].

A. Measurement Model Assessment

The reliability and validity of the measurement model were first assessed using SmartPLS. Three indicators were used to evaluate construct reliability and validity: Cronbach's Alpha, Composite Reliability, and Average Variance Extracted (AVE). Table III presents the results of this assessment [17].

All constructs exceed the recommended thresholds of 0.70 for reliability and 0.50 for AVE, confirming that the measurement model demonstrates strong internal consistency and convergent validity.

TABLE III. MEASUREMENT MODEL

Construct	Cronbach's Alpha	Composite Reliability	AVE
Technological Readiness	0.89	0.92	0.71
Organizational Readiness	0.91	0.93	0.74
Environmental Readiness	0.87	0.90	0.68
Socio-Cultural Readiness	0.88	0.91	0.70

B. Structural Model Analysis

The structural model was evaluated using path coefficients and significance testing to examine the relationships between the readiness constructs and cloud-adoption readiness, as shown in Table IV.

TABLE IV. STRUCTURAL MODEL

Hyp.	Relationship	Path Coeff. (β)	t-value	p-value
H1	Technological Readiness → ISRI	0.36	5.21	< 0.001
H2	Organizational Readiness → ISRI	0.41	6.03	< 0.001
H3	Environmental Readiness → ISRI	0.29	4.38	< 0.001
H4	Socio-Cultural Readiness → ISRI	0.33	4.95	< 0.001

All hypothesized relationships were statistically significant. Among the four readiness factors, organizational readiness had the strongest effect on institutional cloud-adoption readiness, followed by technological readiness and socio-cultural readiness. The model's coefficient of determination ($R^2 = 0.67$) indicates that these four factors jointly explain 67% of the variance in institutional cloud-adoption readiness.

C. Machine Learning Model Performance

The predictive ability of the proposed ISRI framework was assessed using two supervised machine learning approaches, Random Forest and Artificial Neural Network (ANN), trained on 80% of the data and tested on the remaining 20% [17]. Random Forest was chosen for its robustness and ability to handle nonlinear relationships, while ANN was selected for its capacity to model complex interactions among institutional readiness variables. Model performance was measured using accuracy, precision, recall, F1 score, and ROC-AUC, as summarized in Table V.

TABLE V. ML PERFORMANCE

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.91	0.89	0.90	0.89
ANN	0.93	0.91	0.92	0.91

The results indicate that the Artificial Neural Network achieved the highest predictive accuracy, at 93%, outperforming the Random Forest model in capturing nonlinear relationships among institutional readiness variables [7], [18].

Fig. 4 shows the ROC curves for all four models. Consistent with Table V, the Artificial Neural Network achieved the highest area under the curve (AUC = 0.92), followed by Random Forest (AUC = 0.86) and XGBoost (AUC = 0.85), while Logistic Regression recorded the lowest AUC (0.74), confirming that the ANN provides the strongest discrimination among institutional readiness categories [18].

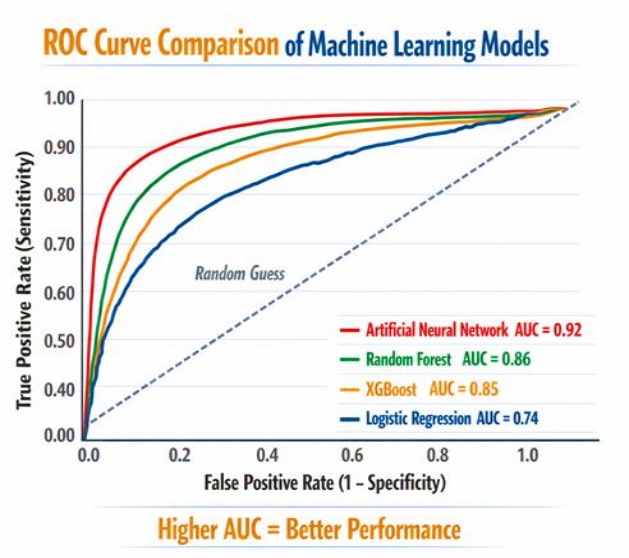


Fig. 4. ROC curve comparison of machine learning models.

D. Ablation Study

To evaluate the contribution of the SEM-derived weighting mechanism and the ISRI formulation, an ablation study was conducted comparing three experimental configurations, as shown in Table VI:

- Baseline ML model (raw features).
- ML model with aggregated ISRI (equal weights).
- Proposed SEM-weighted ISRI + ML model.

TABLE VI. ABLATION STUDY RESULTS

Model Configuration	Accuracy	F1 Score
Raw Features (ML only)	0.86	0.84
ISRI (Equal Weights)	0.89	0.87
SEM-Weighted ISRI (Proposed)	0.93	0.91

The results demonstrate that integrating SEM-derived weights significantly improves predictive performance compared with both the raw-feature model and the equally weighted composite index, confirming that theoretically grounded feature weighting enhances the machine learning model's generalization capability.

E. Confusion Matrix Analysis

A confusion matrix analysis was conducted to evaluate the classification performance of the neural network model across the four readiness categories: Low, Moderate, High, and Strategic readiness [7]. Fig. 5 illustrates the results.

The confusion matrix shows that the neural network model correctly classifies the majority of institutions into their respective readiness categories, with misclassifications occurring mainly between adjacent readiness levels [7].

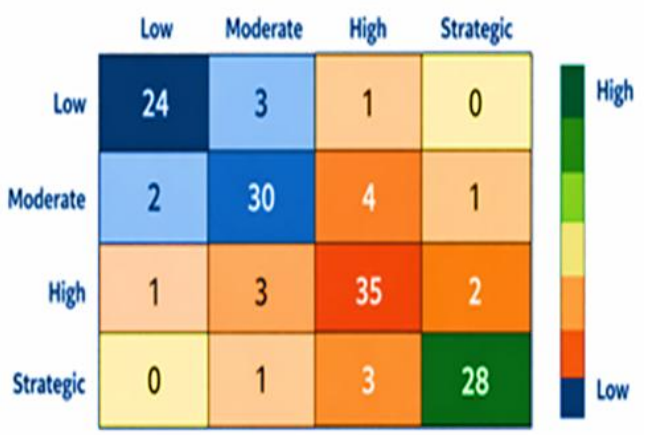


Fig. 5. Confusion matrix for ANN-based readiness classification.

F. ROC Curve and AUC Analysis

Building on the ROC analysis introduced in Section IV-C, an AUC value above 0.90 is generally considered indicative of excellent model-discrimination capability. The Artificial Neural Network's AUC of 0.92 therefore confirms that the proposed predictive framework effectively distinguishes between institutional readiness levels [18].

G. Feature Importance Analysis

Feature importance analysis was performed using the Random Forest model to identify the most influential predictors of cloud-adoption readiness. As shown in Table VII, the resulting scores highlight the relative contribution of each readiness indicator to model performance, identifying the most influential factors in cloud-computing adoption [7].

TABLE VII. FEATURE IMPORTANCE SCORES

Feature	Importance Score
Organizational Readiness	0.34
Technological Readiness	0.29
Socio-Cultural Readiness	0.22
Environmental Readiness	0.15

These results indicate that organizational readiness has the greatest impact on cloud-adoption readiness, followed by technological readiness, which also exerts a considerable influence, particularly in institutions with well-established ICT infrastructure and integration capabilities.

H. Comparative Analysis

To further assess the efficacy of the proposed ISRI model, its predictive performance was compared against traditional statistical prediction models and additional machine learning benchmarks, as shown in Table VIII. The machine learning models performed better overall, as they were better able to capture nonlinear relationships among the institutional readiness factors. Similar hybrid approaches in technology-adoption studies have likewise reported improved prediction accuracy when combining machine learning with structural equation modeling [18], and the resulting predictive analytics support more effective evaluation of institutional readiness for cloud-technology adoption.

TABLE VIII. COMPARATIVE MODEL PERFORMANCE

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.88	0.86	0.87	0.86
XGBoost	0.92	0.90	0.91	0.90
Random Forest	0.91	0.89	0.90	0.89
ANN	0.93	0.91	0.92	0.91

The paired t-test performed on the cross-validation outcomes provides evidence that the performance advantage of the ANN compared to SVM and Random Forest is statistically significant ($p < 0.05$), which increases confidence in the validity of the proposed framework.

In general, the analysis of structural models reveals that each of the four types of readiness considered (technological, organizational, socio-cultural, and environmental) has a statistically significant impact on the level of institutional readiness to adopt cloud computing. Organizational readiness turned out to be the most impactful, emphasizing the significance of leadership, governance, and investments [16], followed by the high impact of technological readiness related to ICT infrastructure, systems integration, and networking capabilities [13]. The same could be said about socio-cultural readiness (which involves digital literacy, innovation culture, and acceptance of users) [14] and environmental readiness (which is manifested through government policy and regulation) [13]. The machine learning algorithms captured such connections in an effective way; both ANN and Random Forest were more capable of modeling these connections compared to the traditional statistical approaches. Embedding SEM-based weights into the ISRI framework helped to overcome the issue of the "black box" problem characteristic of machine learning.

This research adds to the literature by presenting an index of predictive readiness for cloud adoption known as the ISRI, which improves on the previously existing models in [1], [2], [6] and [8], all of which solely take into consideration the technological and organizational aspects. By integrating both explanatory and predictive capabilities, the hybrid approach of SEM and machine learning will assist organizations in understanding the determinants of readiness as well as predicting their future adoption levels [10], [14]. Organizations should give priority to organizational readiness, improve their ICT infrastructure, and ensure digital literacy [4], [11], while policymakers should develop suitable regulatory frameworks [15]. From the institutional point of view, the developed ISRI model gives a practical ground for institutions to determine their readiness level and make improvements before adopting the cloud.

V. CONCLUSION

This study proposed a Machine Learning–Driven Information Systems Readiness Index (ISRI) that integrates Structural Equation Modeling with machine learning by embedding SEM-derived construct weights directly into the feature-engineering process, rather than treating the two approaches as separate analytical stages. Using data from 312 institutional responses across 12 tertiary institutions in Ghana, the structural model confirmed that technological, organizational, environmental, and socio-cultural readiness all

significantly influence institutional readiness for cloud adoption, with organizational readiness emerging as the strongest predictor. The Artificial Neural Network achieved the highest predictive performance, with 93% accuracy and an AUC of 0.92, outperforming Random Forest, XGBoost, Support Vector Machine, and Logistic Regression baselines. The ablation study further confirmed that embedding SEM-derived weights into the feature space produced measurable accuracy gains over both raw-feature and equally weighted models, demonstrating the practical value of theory-driven feature engineering for improving predictive accuracy in intelligent decision-support systems for higher education.

From a practical standpoint, the proposed ISRI framework provides university administrators and policymakers with a theoretically grounded and computationally efficient tool for assessing institutional readiness and prioritizing investment in ICT infrastructure, organizational capacity, and digital-literacy initiatives ahead of cloud adoption.

This study has some limitations. The dataset was drawn from a single national context, Ghana, which may limit the generalizability of the findings to other regions with different institutional and regulatory environments. In addition, the decision support system described in Section III-K remains a conceptual prototype and has not yet been implemented or empirically validated.

Future work should extend the proposed framework to cross-country and cross-regional datasets to test its generalizability, explore more advanced machine learning and deep learning architectures, and design and implement a fully functional, ISRI-based real-time decision-support system with a validated user interface. Longitudinal studies that track institutional readiness and actual cloud-adoption outcomes over time would also help establish the predictive validity of the ISRI framework beyond cross-sectional analysis.

REFERENCES

[1] E. D. Armah and I. S. Ali, "A comprehensive analysis of faculty adoption of cloud computing e-learning in Ghanaian technical universities," Int. J. Eng. Pedagogy, vol. 14, no. 5, pp. 65–87, 2024, doi: 10.3991/ijep.v14i5.47841.

[2] E. K. Agbemafle, B. B. Benuwa, and B. Ghansah, "Enhanced technology acceptance model for evaluating adoption of cloud computing in colleges of education in Ghana," in Proc. 2024 IEEE SmartBlock4Africa Conf., 2024, doi: 10.1109/SmartBlock4Africa61928.2024.10779531.

[3] A. S. Mohammed, C. Amoah, and I. B. Abdulai, "Technology integration on infrastructure to support educational needs in a college in Ghana," Infrastruct. Asset Manag., early access, 2025, doi: 10.1680/jinam.24.00030.

[4] D. K. Dake, G. K. Bada, and A. E. Dadzie, "Internet of things (IoT) applications in education: Benefits and implementation challenges in Ghanaian tertiary institutions," J. Inf. Technol. Educ.: Res., vol. 22, pp. 311–338, 2023, doi: 10.28945/5183.

[5] A. S. Yakubu and A. M. Kassim, "Exploring the empirical studies of cloud computing adoption in anglophone West African countries' academic libraries: A review," in Proc. SPIE, vol. TBD (article no. 2675083), 2023, doi: 10.1117/12.2675083.

[6] M. A. Adarkwah and R. Huang, "Technology addiction, abduction and adoption in higher education: Bird's-eye view of the ICT4AD policy in Ghana 20 years on," Br. J. Educ. Technol., vol. 54, no. 6, pp. 1458–1476, 2023, doi: 10.1111/bjet.13352.

[7] R. B. Boison, "Barriers and enablers to artificial intelligence (AI) adoption in administrative functions in public universities in Ghana: A case study of the University of Education, Winneba," Can. J. Educ. Soc. Stud., vol. 5, no. 4, pp. 1–22, 2025, doi: 10.53103/cjess.v5i4.377.

[8] A. Mwotil, B. Kanagwa, A. Zawedde, and T. E. Anderson, "Cloud adoption in low-resource settings: A case study of higher education institutions in Uganda," in Proc. Int. Conf. e-Infrastructure and e-Services for Developing Countries, Cham, Switzerland: Springer, 2023, pp. 15–28, doi: 10.1007/978-3-031-81570-6_2.

[9] P. K. Adjei, T. M. Gadafi, and Z. Qin, "Enhancing business intelligence in Ghana through digital technologies: Optimizing management information sharing with cloud computing, big data, and AI," in Proc. 2025 Int. Conf. Digit. Transform. Innov., New York, NY, USA: ACM, 2025, doi: 10.1145/3764727.3764770.

[10] J. K. Adjei, S. Adams, and L. Mamattah, "Cloud computing adoption in Ghana: Accounting for institutional factors," Technol. Soc., vol. 65, Dec. 2021, Art no. 101583, doi: 10.1016/j.techsoc.2021.101583.

[11] E. D. Armah and I. S. Ali, "Navigating anxiety in a cloud e-learning virtual environments: The moderating role of anxiety on lecturers' adoption of cloud computing e-learning," Int. J. Interact. Mob. Technol., vol. 18, no. 17, pp. 4–20, 2024, doi: 10.3991/ijim.v18i17.49123.

[12] J. K. Boateng, "Managing learning outcomes with technology in Ghanaian higher education," Cogent Soc. Sci., vol. 9, no. 1, Dec. 2023, Art no. 2282507, doi: 10.1080/23311886.2023.2282507.

[13] R. Y. Kudozia, A.-R. Salifu, D. Owusu-Donkor, and R. Gyane, "Cloud readiness and digital infrastructure gaps in Ghana," Int. J. Res. Sci. Innov., vol. 12, no. 11, pp. 2053–2086, 2025.

[14] A. A. Afuwape, "Analysis of cloud computing implementation in higher education: The case of selected West African universities," M.S. thesis, Univ. Eastern Finland, Joensuu, Finland, 2022. [Online]. Available: http://erepo.uef.fi/handle/123456789/

[15] A. Al-Mousa, "Cloud computing: Bridging the link between industry and the classroom," Int. J. Electr. Eng. Educ., 2022, doi: 10.1177/0020720919857636.

[16] E. Entsie et al., "Assessing the mediating role of top management support in the adoption of cloud computing in the public sector: The case of Ghana," Manag. Sci. J., 2025.

[17] K. K. Hiran et al., "Predicting the core determinants of cloud-edge computing adoption (CECA) for sustainable development in the higher education institutions of Africa: A high order SEM-ANN analytical approach," Technol. Forecast. Soc. Change, vol. 199, Jan. 2024, Art no. 123664, doi: 10.1016/j.techfore.2023.123664.

[18] A. Rehaimi et al., "Towards a federated and hybrid cloud computing environment for sustainable and effective provisioning of cyber security virtual laboratories," Expert Syst. Appl., 2024, doi: 10.1016/j.eswa.2024.124267.

[19] G. C. Yamson, B. O. Akoto, E. D. Yamson, and A. B. Donkor, "Online learning in the new normal: University students' perceptions, experiences and challenges in the post-COVID-19 era," Issues Trends Learn. Technol., vol. 13, no. 1, pp. 70–92, 2025.

[20] H. Rajora, D. Jeffery, and A. Amoozegar, "Automating corporate strategy: How cloud AI transforms managerial intelligence," in Automating Intelligence With Cloud-Native AI Tools. Hershey, PA, USA: IGI Global, 2026, pp. 183–195, doi: 10.4018/979-8-3373-7442-0.ch013.

Color and Emotion in User Experience Design for Older Adults' Health Applications: A Multi-Database Bibliometric and Evidence-Mapping Study

YangFan He[1], Azizah Che Omar[2]*, Sobihatun Nur Abdul Salam[3]
School of Multimedia Technology and Communication, Universiti Utara Malaysia, Sintok, Kedah, Malaysia[1, 2, 3]
Fujian Forestry Vocational & Technical College, Nanping, Fujian, China[1]

***Abstract*—Color is routinely present in health applications, but evidence about its role in older adults' user experience remains dispersed across design, health informatics, and human-computer interaction. This five-database study maps an affect-or-color focal corpus and applies a structured design-evidence audit to its color-term candidates. Searches of Web of Science Core Collection, Scopus, PubMed, IEEE Xplore, and the ACM Digital Library produced 2,515 source rows, 1,760 unique documents, and 941 focal records. A record-level audit of the classifier defining that corpus returned precision 0.880 and recall 0.982. Performance indicators, knowledge networks, and temporal bursts describe the corpus, and their stability was tested against retrieval, threshold, and periodization choices. Two design experts independently coded 41 color-term candidates, using full text for 23 records. Agreement reached 95.1% for inclusion (Cohen's kappa = 0.918) and 90.2% for population relevance (kappa = 0.853). Reconciliation retained ten primary older-adult studies, three mixed-age studies, and one contextual review; 27 records were excluded, and five of the ten color-and-emotion bridge records met the primary rule. Across the primary set, color supported legibility, risk communication, hierarchy, navigation, aesthetic preference, and affective tone, but most studies bundled color with other interface features. Preference-to-performance evidence was absent, and safety-relevant findings stayed embedded in broader workflows. The evidence map separates reported design functions from causal claims and identifies priorities for controlled validation.**

***Keywords*—Older adults; health applications; user experience; interaction design; color; emotion; bibliometrics; science mapping**

I. Introduction

Health applications mediate monitoring, consultation, education, and self-management for many older adults. Value rests on more than clinical content. Changes in vision, cognition, motor control, health literacy, and technology experience affect how information is found and acted upon. Wildenbos et al. locate these barriers within a mobile-health usability framework [1]. Acceptance studies add usefulness, compatibility, trust, and prior experience [2], [3]. Age-inclusive interaction design is therefore a task and evidence problem, not a matter of enlarging controls.

Color complicates that problem. It may guide attention, mark a status, organize a display, or alter the felt character of an interface. Its effect is conditional. Lightness and saturation relate to pleasure and arousal in controlled studies [7], [8].

Ecological and cross-cultural research emphasizes learned association and object context [9], [10], [11]. A palette that is liked cannot be assumed to improve comprehension, safe action, or continued use in a health task.

The relevant scholarship is distributed across health informatics, human-computer interaction, gerontology, medicine, service research, and design. Existing reviews usually remain inside one domain. They seldom connect publication growth and collaboration with the smaller body of studies that actually report color as an interface variable. A narrow conjunctive search would create the opposite problem by discarding the user-experience literature in which color questions are embedded. The gap is thus architectural as well as substantive.

This study addresses that gap with a two-layer corpus in which each layer keeps its own denominator. The outer layer is a focal map of 941 records. It describes production, collaboration, vocabulary, intellectual sources, and temporal fronts, and its task is to locate color work inside the surrounding user-experience literature rather than to supply design evidence. The inner layer is a structured design-evidence matrix. It examines 41 color-term candidates for age relevance, interface function, health task, perceptual or affective mechanism, behavioral outcome, and safety relevance, and it resolves to the 14 records that carry the study-level claims. The contribution lies in joining the two layers while holding their denominators apart, so that corpus-level visibility is never read as participant-level support. The analysis stops short of estimating clinical effectiveness and of proposing a universal palette. Four research questions organize the study.

RQ1: How have publication volume, source concentration, authorship, institutional participation, and country output developed in the retrieved field?

RQ2: Which collaboration, keyword, co-citation, and bibliographic-coupling structures organize its social, conceptual, and intellectual base?

RQ3: Which themes and citation fronts emerged or persisted across the study period?

RQ4: Within the color-term candidate set, which interface functions, health tasks, affective constructs, behavioral outcomes, attribution limits, and safety concerns are documented?

*Corresponding author.

II. Related Work and Analytical Position

A. Older-Adult User Experience in Health Applications

Older adults are not a uniform user group: functional ability, health literacy, device experience, and motivation all vary within the population. Studies of eHealth adoption therefore place usefulness, support, and trust alongside usability [2], [3], and digital literacy further conditions trust in online health information [19]. Older patients describe late-life learning demands as a barrier to eHealth use [20]. Interaction quality is relational: a visual feature acquires meaning through the task, the setting, and the person who encounters it.

Measurement remains inconsistent. Engagement reviews report many custom indicators and limited comparability across health applications [5]. Validated instruments such as uMARS, MAUQ, and the System Usability Scale provide more stable anchors [21], [22], [23], yet they leave color mechanisms unresolved, and a favorable usability score can coexist with inaccessible contrast or ambiguous alert semantics. The design question is more specific: which visual and interaction properties support a defined task for a defined user group?

B. Color, Emotion, and Affective Interaction Design

Color may distinguish controls, group information, signal status, reinforce hierarchy, support personalization, or establish an affective tone. These functions need separate treatment. Color-emotion relationships vary with lightness, saturation, object meaning, viewing conditions, and prior association [7], [8], [9], [10], [11]. Health interfaces add a semantic safety constraint. Warning, risk, and medication status should not depend on hue alone. The bibliometric question is therefore whether color is reported precisely enough to support cumulative user-experience knowledge.

C. Why a Bibliometric Map Is Necessary

Bibliometric analysis complements study-level review by revealing production, collaboration, conceptual vocabulary, and citation structure: co-word analysis locates recurring design problems, co-citation mapping identifies the intellectual base, and bibliographic coupling shows which current documents draw on similar references. Three evidence terms are kept distinct throughout. A metadata gap means a field is absent from the searchable export. A reporting gap means full-text verification finds the item missing from the study. An evidence gap means the available outcome evidence falls short of a design conclusion. A dense network testifies to citation and indexing behavior alone.

III. Data and Methods

A. Retrieval Design and Source Boundaries

Searches were executed on 31 July 2026 in Web of Science Core Collection, Scopus, PubMed, IEEE Xplore, and the ACM Digital Library. A single four-block Boolean expression operationalized the conceptual domains of older adults, health applications, user experience and interaction design, and color or affective terms. Every provider received the same four blocks, translated into its own field syntax. What differed between providers was field scope rather than concept coverage: Web of Science matched on Topic, Scopus on title, abstract, and keywords, IEEE Xplore on all metadata, PubMed on title and abstract, and ACM on abstract alone. Truncation appeared only in the fourth block, where emotion* and affect* were expanded by each provider's own stemmer. No date, language, or document-type limit was applied, so the retrieved set spans 2002 to 2026 and includes articles, reviews, and conference papers. Table I records the field scope, limit settings, and export boundary for each database, and Supplementary Material S2 reproduces the five expressions verbatim together with their result counts. A sensitivity analysis repeated the annual, source, keyword, and affiliation-country summaries after the nine PubMed-only records were removed. The study is a bibliometric corpus analysis with PRISMA-informed provenance rather than a systematic review of intervention effects [12].

TABLE I. Database Retrieval and Acquisition Boundaries

Database	Raw rows	Field scope, limits, and capture boundary
Web of Science Core Collection	1,308	Topic field; four blocks; stemmer truncation in block 4; no date, language, or type limit; executed 31 Jul 2026; full records with cited references
Scopus	687	Title, abstract, keywords; four blocks; stemmer truncation in block 4; no limits; executed 31 Jul 2026; result-list metadata
PubMed	356	Title/Abstract tags on all four blocks; stemmer truncation in block 4; no limits; executed 31 Jul 2026; XML and JSON
IEEE Xplore	118	All metadata; four blocks; stemmer truncation in block 4; no limits; executed 31 Jul 2026; result-list metadata
ACM Digital Library	46	Abstract field; four blocks; stemmer truncation in block 4; no limits; executed 31 Jul 2026; result-list metadata

Table I was reconstructed from the five saved result sets and their acquisition logs. Query scope appears beside export scope because those boundaries determine which fields can be pooled. All five providers ran the same four conceptual blocks; PubMed applied them within title and abstract, which narrows field scope without narrowing the concept set. Export depth differed more than query scope. Web of Science supplied full records and cited references, and PubMed supplied structured XML and JSON. Scopus, IEEE Xplore, and ACM records came largely from provider result lists, so abstracts, affiliations, keywords, and cited references were unevenly available. Table I therefore defines an analytical boundary rather than a list of interchangeable database samples.

Fig. 1 traces the provenance chain from the five database result sets to the study-level evidence tiers, following 2,515 source rows through 1,760 unique documents, 941 focal documents, and 41 color-term candidates to the resolved evidence tiers. Each stage carries a distinct denominator, which keeps metadata term occurrence from being mistaken for verified participant evidence.

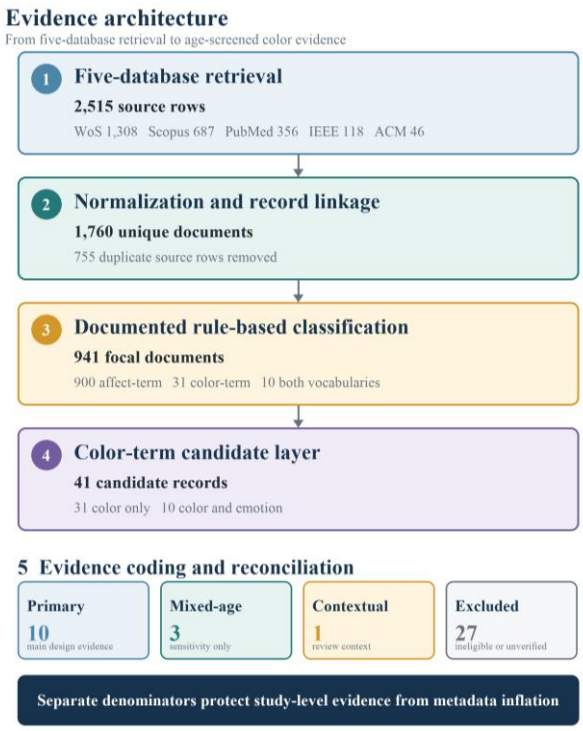


Fig. 1. Evidence architecture.

B. Normalization, Record Linkage, and Corpus Definition

OpenRefine 3.10.1 recorded 21 operations for whitespace, Unicode, identifiers, source labels, and document fields. DOI strings were lower-cased, stripped of resolver prefixes, and checked against title, year, and author evidence. Exact normalized titles were linked only when publication years differed by no more than one year. Unresolved candidates were checked against Crossref. A candidate was accepted only when title similarity reached 0.92, the year was compatible, and first-author evidence did not contradict it. The result contained 1,760 unique documents after 755 duplicate source rows were removed. Database coverage was one source for 1,315 documents, two sources for 140, and three sources for 305.

A documented rule-based classifier read titles, abstracts, author keywords, and index keywords. Color terms covered color, colour, chromatic, hue, lightness, brightness, palette, and color-qualified contrast or saturation. The affective vocabulary included emotion, affect, valence, arousal, trust, comfort, anxiety, enjoyment, frustration, fear, calmness, and satisfaction. The classifier found 900 records containing affect-related terms, 31 containing color-related terms, and 10 containing both vocabularies within the 941-record focal corpus. These labels describe metadata occurrence, not substantive research focus or complete article content. Age relevance was never inferred from a color term, an application name, or a demographic phrase such as people of color.

The classifier was then audited against the locked vocabulary across all 1,760 records rather than a sample, confirming the sense of each matched term and screening excluded records for terms the pattern could not reach. It returned 828 true positives, 113 false positives, 15 false negatives, and 804 true negatives: precision 0.880, recall 0.982, F1 0.928, and Cohen's kappa 0.855. Both error classes are systematic. Every false positive came from the verb affect, as in factors that affect adherence, which a lexical pattern cannot separate from the psychological construct; every false negative came from a morphological variant outside the word-boundary pattern.

Two consequences follow. The color stream is unaffected, since all 41 color-term candidates were confirmed and both error classes fall inside the surrounding affect vocabulary. The descriptive map also holds when the false positives are removed, retaining the ten leading sources and nineteen of the twenty leading keywords at rank correlations of 0.981 and 0.988, with the annual series at 0.995. The focal corpus is therefore an inclusive frame whose affect margin is documented rather than hidden (S5).

A further check addressed whether predefined terminology misses studies that describe color indirectly. The retrieval blocks already admit contrast, brightness, saturation, palette, trust, and engagement, so indirect descriptions enter at the search stage; the classifier vocabulary is narrower, requiring a color-qualified reading of contrast and saturation. An expanded probe adding unqualified contrast, visualization, display, theme, aesthetic preference, legibility, luminance, color-vision terms, named hues, and visual highlighting surfaced 546 further records, of which 111 survived sense disambiguation. None reported color as a manipulated or evaluated interface variable; the descriptors resolved to thematic-analysis themes, app-rating aesthetics subscales, data-dashboard visualization, and rhetorical uses. Seven records were kept for full-text verification (S2).

C. Bibliometric and Network Analyses

Bibliometrix 5.4.1 and Biblioshiny supported annual production, source and author-name-string productivity, thematic mapping, and thematic evolution [13]. VOSviewer 1.6.21 generated author, institution, and country collaboration maps, keyword co-occurrence, reference co-citation, and document-coupling networks [14], [15]. Fractional counting informed interpretation where large teams could otherwise dominate links [16]. Affiliation countries were parsed from all available address strings. Institution labels were case-folded and reconciled with a documented alias table; no ORCID-based or registry-based entity resolution was available.

TABLE II. NETWORK OBJECTS AND RETAINED ANALYTICAL MATRICES

Object	Nodes	Undirected edges
Author collaboration	250	663
Institution collaboration	250	547
Country collaboration	72	346
Keyword co-occurrence	250	11,494
Reference co-citation	300	6,427
Document coupling	941; 838 with cited references	20,161

Table II was generated from the retained adjacency matrices rather than copied from network images, and identifies the six network objects with their node and edge denominators. All six used full counting, association-strength normalization, and VOS clustering at resolution 1.00. Edge weight is the raw co-

occurrence, co-citation, or shared-reference count before normalization; nodes entered by rank on total link strength with occurrence weight as tie-breaker; isolates were dropped after truncation; and display subsets were pruned separately for single-column readability. Node size represents production or occurrence, line width link weight, and color the VOS cluster. Address data were available for 861 documents, affiliation-country fields for 855, institution fields for 839, and cited references for 838, so although the coupling matrix preserves all 941 rows, only those 838 records contribute nonzero coupling. S3 lists every parameter and retained node count by network.

Because structure responds to these choices, the six maps were re-clustered under varied parameters and compared with the published solution by adjusted Rand index, moving resolution between 0.7 and 1.3, substituting raw for association-strength weighting, and dropping the weakest quartile of nodes. Central nodes held throughout: the ten highest-strength nodes were retained in every configuration of all six networks. Cluster boundaries proved more sensitive and tracked density, holding at 0.80 to 1.00 for the sparse author and institution maps but ranging from 0.28 to 0.70 for the denser keyword, co-citation, coupling, and country maps, with community counts varying between two and eight. Semantic cluster membership is therefore reported as one defensible partition rather than a unique solution, with the interpretive weight on central nodes (S4).

D. Temporal Fronts and the Color-Explicit Design Matrix

Thematic evolution used three descriptive publication phases. Counted over complete years only, these hold 60 documents for 2009 to 2017, 267 for 2018 to 2022, and 437 for 2023 to 2025. The 177 records retrieved from January to July 2026 are reported separately as a partial observation and are excluded from every trend statement. The cut points mark the first year above 20 annual records and the later rise above 80, rather than equal-duration or equal-volume windows.

Because publication volume is a weak basis for claims about stages of development, both boundaries were tested against change points in the 2009 to 2025 series. A Pettitt test located a single change point at 2017 ($K = 72$, $p = 0.005$), adjacent to the first boundary. Two-breakpoint segmentation recovered 2023 on the raw-count scale, where the 2018 and 2023 pair ranked third among admissible pairs, but on a logarithmic scale the first boundary moved to 2012 or 2014, because the sparse years from 2009 to 2013 dominate that transformation. The second boundary is thus reproducible and the first scale-sensitive, so the three phases serve as descriptive reporting windows whose thematic labels summarize each window rather than mark a detected transition.

Keyword and cited-reference bursts were detected with Kleinberg's finite-state model [17]. The scale parameter was 2, the transition penalty was 1, the minimum keyword document count was 8, and the minimum cited-reference count was 5. DOI variants were merged before modeling. The procedure retained 198 keyword candidates and 286 cited-reference candidates. Because 2026 was incomplete, burst detection ended in 2025; the temporal corpus contained 764 documents, including 685 with cited references.

Two design experts independently coded all 41 color-term candidates using separate worksheets and a locked codebook. Fields covered population relevance, inclusion status, interface element, health task, color operation, proposed mechanism, affective construct, outcome, attribution, safety relevance, evidence location, and access status. Primary evidence required an explicit older-adult designation supported by participant age information or an older-adult user test, a health-related digital interface, and explicit color evidence. A record was mixed-age when older participants or a documented older subgroup were present without a separate older-adult result. Target-only expert studies and reviews were classified as contextual rather than primary. Inclusion decisions agreed for 39 of 41 records (95.1%; Cohen's kappa = 0.918; bootstrap 95% CI 0.80 to 1.00). Population relevance agreed for 37 of 41 (90.2%; kappa = 0.853; 95% CI 0.70 to 0.97). A third reviewer reconciled six records containing a disagreement in either field against the most complete accessible source. Secondary categorical kappa values ranged from 0.705 to 1.000; overlapping open-text design fields remained descriptive.

E. Quality Controls and Reproducibility

Quality control was applied at the source, record, corpus, and result levels. Source counts were reconciled against the five saved result sets before parsing, and each raw row retained its database name, provider identifier, source file, and URL. Field-completeness audits were run separately for DOI, title, author, abstract, keyword, affiliation, citation, and cited-reference data, so missingness was visible before any network was built. Optional fields had no bearing on corpus membership. Analytical outputs were regenerated from the retained corpus rather than transcribed from images, and each figure is linked to its map, network, or tabular source file.

The linkage audit recorded every DOI union and exact-title union, together with the Crossref score and DOI origin. A resolver check identified the provider DOI 10.2024/1/e52442 in cluster CE-00953 as non-resolving. Title, author, year, and Crossref metadata confirmed the canonical DOI 10.2196/52442, which is the identifier used for interpretation in the corrected audit. This cluster contains four source rows from Web of Science, Scopus, and PubMed but represents one document. It is not a legitimate reprint and is not counted twice.

One interpretive boundary was fixed before the maps were read. Node prominence measures bibliometric visibility, and its warrant stops there; design and clinical effectiveness lie outside what it can register. Together with the gap vocabulary of Section II-C, this keeps the quantitative map within the limits of its evidence.

IV. Results

The results follow the two layers of the corpus. Sections IV-A to IV-D describe the 941-record focal map and the user-experience literature surrounding color work, each stating what its finding implies for the color layer; Section IV-E then resolves that layer from the 41 color-term candidates to the 14 records carrying study-level evidence. Denominators differ at every step and are named wherever a count appears.

A. Corpus Formation and Growth

The 2,515 source rows contracted to 1,760 unique documents, a retention rate of 69.98%. Within the corpus, 1,311 documents came from one raw source row, 144 from two, 304 from three, and one from four; this differs from the database-coverage distribution in Section III-B because some documents were duplicated inside a single database. The CE-00953 DOI was canonicalized to 10.2196/52442 after resolver and Crossref checks, and no document was retained as a separate reprint on the basis of the invalid provider DOI.

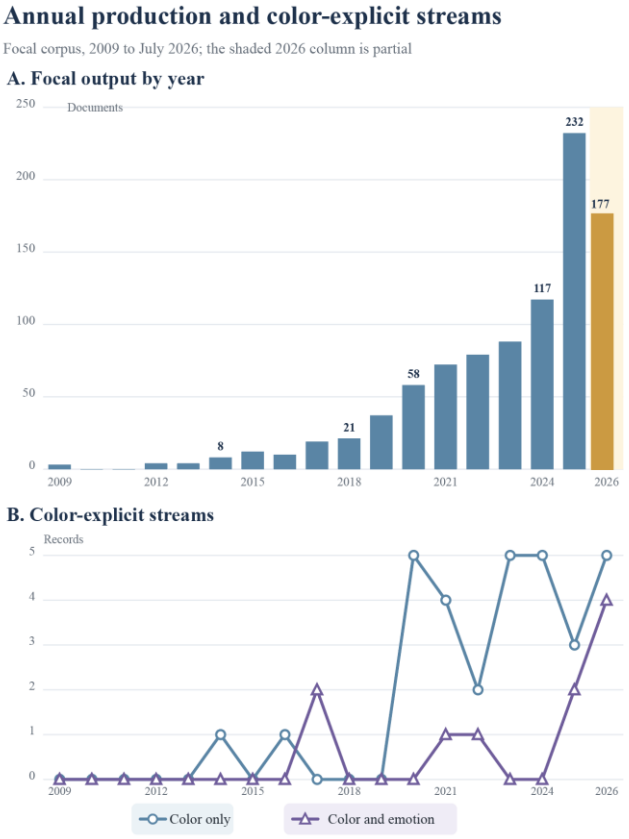


Fig. 2. Annual production and color-term streams.

Fig. 2 comes from the deduplicated annual-production table. Panel A shows the scale of the focal corpus; Panel B retains the smaller color-term streams on their own axis. Annual focal output rose from 21 documents in 2018 to 58 in 2020, 117 in 2024, and 232 in 2025. The 177 records retrieved from January to July 2026 cover seven months and are shaded as a partial observation. Every month, they run at 25.3 records against 19.3 for 2025, which annualizes to roughly 303. That comparison shows only that the partial year is not evidence of decline; because submission and indexing are uneven within a year, the 2026 figure stays out of every growth statement and out of burst detection. Excluding nine PubMed-only records left 932 documents and changed any annual count by no more than three. The top ten source set and top twenty keyword set were unchanged; frequency rank correlations were 0.980 and 0.994, respectively. Nine of the ten leading affiliation countries remained, with rho = 0.996. The sensitivity result supports the stability of the descriptive map without making the PubMed query prevalence-equivalent.

B. Sources, Contributors, and Collaboration

Case-folding and alias inspection reduced 407 nonblank source labels to 399 normalized labels. JMIR Formative Research contributed 101 focal documents. The Journal of Medical Internet Research contributed 63, JMIR Research Protocols 41, JMIR mHealth and uHealth 36, JMIR Human Factors 35, and both Digital Health and JMIR Aging 26. The source core belongs mainly to digital health and health informatics. General HCI and design venues form a smaller visible layer, although the retrieval design cannot establish what lies beyond the corpus.

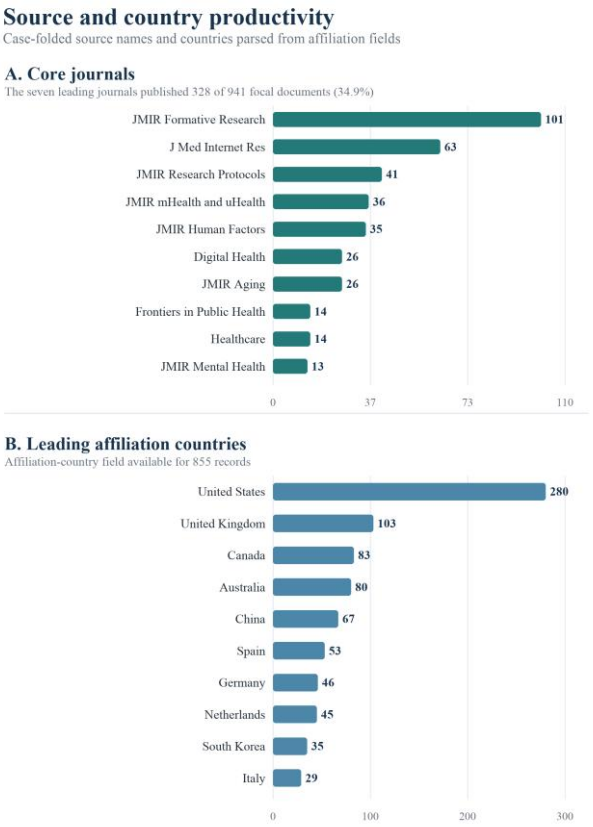


Fig. 3. Core journals and leading affiliation countries.

Fig. 3 places source concentration beside country participation parsed from affiliation addresses. The panels use different metadata fields. The United States appeared in the affiliations of 280 documents, followed by the United Kingdom in 103, Canada in 83, Australia in 80, and China in 67; these are affiliation-presence rather than corresponding-author counts. Institution strings were normalized separately. The University of Toronto led with 19 documents, the University of Sydney with 14, and the University of British Columbia with 13. These values describe recorded participation, whereas Fig. 4 examines collaboration strength.

After source-name normalization, the seven leading sources accounted for 328 documents, or 34.86% of the focal corpus, while 323 of the 399 normalized sources contributed one document. Authorship was more dispersed: 6,969 normalized author-name strings were recorded, of which 6,722 appeared once. Institution strings showed a similar long tail, with 2,014 of

2,262 occurring in one document. The pattern indicates dispersion at the recorded-name level.

Those institution counts rest on string identity, so the leading entities were audited against the address field, which exposed two opposite failures. Splitting is the larger: 1,076 of the 2,262 institution strings retain a fragment of the bracketed author list preceding each Web of Science address, so one institution can appear under several labels. Re-parsing and crediting each document once raises the leading counts to 26, 22, and 20 for Toronto, Sydney, and British Columbia without changing their order. Merging is the opposite risk, since a token-insensitive normalization would combine Univ Washington in Seattle with Washington Univ in St. Louis, which the city field separates. Across the published top thirty, the two parses correlate at rho = 0.759 and seven of ten identities persist, so the three leading institutions are stable while positions below them turn on identity decisions string matching cannot settle (S6).

This distribution bears directly on the color layer. A concentrated core coexists with a long tail of single-study contributors, and across that tail color surfaces under varied labels: contrast, theme, display, visualization, palette, or aesthetic preference. The vocabulary probe reported in Section III measured that dispersion, finding 111 records that use such descriptors while carrying no color term in their metadata. Section V returns to what the pattern implies for reporting practice.

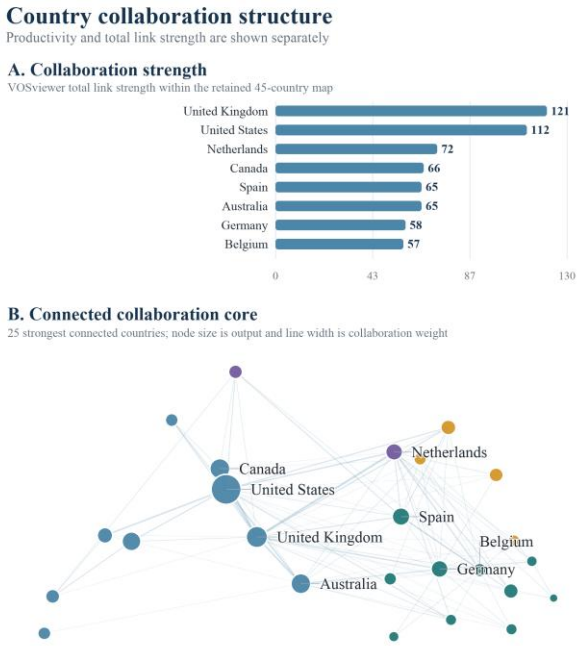


Fig. 4. Country productivity and collaboration structure.

Fig. 4 separates publication output from network connection. Panel A reports total link strength in the retained 45-country VOSviewer map; Panel B shows the 25 strongest connected countries and their 100 strongest links. The United Kingdom had the highest total link strength at 121, followed by the United States at 112. The Netherlands reached 72, Canada 66, Spain and Australia 65 each, Germany 58, and Belgium 57. The University of Toronto led the institutional network with total link strength 41, while Bucci and Schwannauer each reached 64 in the author network. These values refer only to the retained maps and available affiliation fields. Having distinguished productivity from connection, the analysis turns to the vocabulary that structures the field.

C. Conceptual Structure and Thematic Evolution

Raw author-keyword frequencies were led by mHealth with 199 documents, digital health with 174, mobile health with 118, usability with 111, eHealth with 109, and older adults with 95. Mental health, depression, anxiety, telehealth, acceptance, user experience, and feasibility formed the next layer. Color terms remained outside the dominant raw vocabulary. Because mHealth and mobile health, as well as eHealth and digital health, are related but not identical labels, the analysis reports them separately and interprets their combined design meaning cautiously.

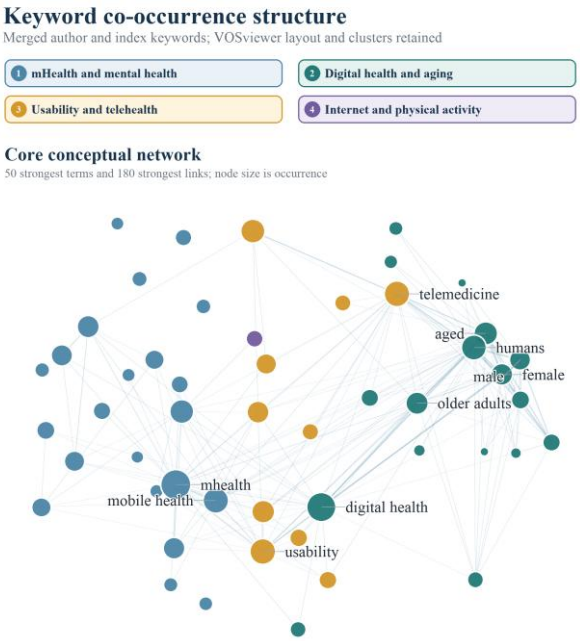


Fig. 5. Keyword co-occurrence structure with named conceptual clusters.

Fig. 5 was reconstructed from the saved VOSviewer map using the 50 terms with the greatest total link strength and the 180 strongest links. Four communities organize the visible vocabulary. The first connects mHealth and mobile health with mental health, depression, anxiety, and intervention. The second links digital health with older adults and demographic indexing. A third community brings usability, telemedicine, self-management, and acceptance together. The smaller fourth group concerns internet use and physical activity. Cluster names above the network state the interpretation in words, reducing reliance on color alone. Color terminology again stays outside the dominant keyword core.

The thematic map refines this reading without assigning causal status to clusters. The eHealth, depression, and anxiety cluster has the highest relative centrality and density; mHealth and usability are central but less internally specialized; digital health and aging show developing connectivity; and telehealth

care occupies a less central, less dense position. These positions describe how terms are organized in the retrieved literature, leaving effectiveness to study-level evidence.

Thematic evolution shows a change in the vocabulary used by the retrieved literature. From 2009 to 2017, labels centered on technology, telemedicine, mobile health, telehealth, and intervention. During 2018 to 2022, the vocabulary consolidated around mHealth, telemedicine, and anxiety. From 2023 onward, digital health and usability became more explicit destination themes. Word-flow links early telemedicine to later digital-health and implementation vocabulary. The pattern marks greater visibility of interaction-quality terms, not evidence of theoretical maturation.

D. Intellectual Base and Current Coupling Fronts

The co-citation base combines usability measurement, technology acceptance, clinical screening, research infrastructure, and digital-mental-health guidance. Representative references include the System Usability Scale [23], the Unified Theory of Acceptance and Use of Technology [24], GAD-7 [25], REDCap [26], and work on engagement with mental-health applications [27]. These references form the methodological backbone of the map. They show that the field draws on HCI, information systems, psychometrics, and health informatics rather than on one established color-design tradition.

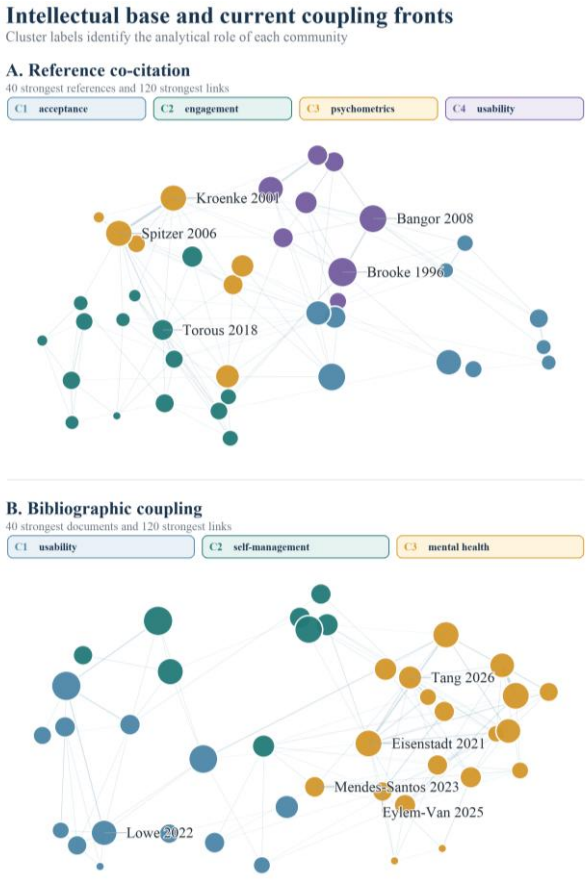


Fig. 6. Co-citation and document-coupling structures.

Fig. 6 distinguishes the intellectual base from current coupling fronts. Panel A contains four co-citation communities centered on acceptance, engagement, psychometrics, and usability. Representative references include UTAUT, GAD-7, engagement guidance, and the System Usability Scale [23], [24], [25], [27]. Panel B contains three displayed document-coupling communities concerned with usability, self-management, and mental health. Short labels identify the documents actually retained in the single-column map. Telehealth satisfaction and application-based physical-activity research belong to the wider coupling corpus [28], [29], but are not among its displayed labels. The structure is grounded in measurement and adoption rather than a mature color-design tradition.

Citation counts provide a complementary snapshot, taken as the maximum provider count in the saved July 2026 exports. Donker et al.'s systematic review of mental-health applications led with 774 citations [4]. Li et al.'s older-adult wearable acceptance model followed with 328, Lehtimaki et al.'s digital-mental-health overview with 302, and Ng et al.'s engagement review with 200 [2], [5], [6]. Fan et al.'s real-world chatbot study also ranked highly [18]. Being neither field-normalized nor cross-provider equivalent, these values support ranking only within the assembled corpus.

Co-citation and coupling answer different questions. The co-citation map reflects references that later papers repeatedly place together, so its clusters indicate a shared intellectual base. Coupling depends on the reference lists of current documents and responds more quickly to new problems. The recent coupling front contains a visible 2024 to 2026 layer even though those records have had less time to accumulate citations. That layer sits apart from the historical base in the map, a separation of reference profiles rather than a measure of influence.

E. Temporal Bursts and the Color-Explicit Design Subset

Sixteen keyword burst intervals and seven DOI-normalized cited-reference bursts were detected. Early keyword bursts were broad and persistent: intervention ran from 2009 to 2020 with strength 6.23, internet from 2009 to 2017 with strength 4.47, and technology from 2012 to 2016 with strength 4.22. Later bursts became more methodological. Validation ran from 2020 to 2023 with strength 5.83, while mobile phone ran from 2021 to 2023 with strength 5.25. Female, male, artificial intelligence, and qualitative research appeared as short bursts in 2025. These are indexing and publication fronts, not direct evidence of design effectiveness.

The normalized reference bursts trace a parallel methodological shift. A mobile application rating reference peaked in 2022 [21]. GAD-7 burst from 2020 to 2022, and the digital mental-health engagement review from 2021 to 2022 [25], [27]. REDCap rose in 2023 and 2024, showing increased citation visibility for standardized data infrastructure [26]. UTAUT produced a 2024 burst, marking renewed attention to adoption theory [24]. Two shorter bursts concerned application assessment and public-health implementation. The System Usability Scale did not remain a burst after DOI variants were collapsed, correcting the earlier duplicate-sensitive result.

The burst sequence moves through three emphases. Early fronts concerned whether mobile technology could deliver an intervention. Middle-period fronts emphasized acceptability, psychometrics, and validation. The latest fronts concern how

digital services are implemented and experienced by more specific user groups. Color formed no independent burst at any stage, and its terms stayed embedded in usability, visualization, accessibility, and experience studies.



Fig. 7. Temporal fronts and age-screened color evidence.

Fig. 7 brings temporal fronts and design evidence into one view while keeping their denominators separate. Panel A plots the ten strongest keyword bursts from the complete-year corpus through 2025. Panel B shows the evidence tiers derived from 41 color-term candidates, followed by five nonexclusive characteristics of the ten primary records: full-text availability, color isolation, direct safety relevance, preference-to-performance linkage, and isolated safety endpoints. The panel is an evidence-boundary summary rather than a causal-effect display.

TABLE III. AGE RELEVANCE AND EVIDENCE TIERS IN THE COLOR TERM CANDIDATE SET

Boundary	Result and interpretation
Primary older-adult	10 (6 full text; 4 abstract): main design evidence
Mixed-age evidence	3 (2 full text; 1 abstract): no separate older result
Contextual	1 abstract-level scoping review
Excluded	27 of 41: 24 noneligible, 2 contextual-only, 1 unverified
Primary bridge	5 of 10; no fully isolated color-only effect

Table III formalizes the boundary shown in Fig. 7. Ten records met the primary older-adult criterion, including six coded from full text and four from the most complete abstract-level record. Three mixed-age studies entered sensitivity interpretation; two were supported by full text and one by abstract-level evidence. One abstract-level scoping review remained contextual. The other 27 records comprised 24 noneligible studies, two target-only contextual exclusions, and one unverified title-level record. This tiering prevents the 41-record candidate layer from being interpreted as a homogeneous older-adult evidence set. Study-level fields for all 14 retained records are provided in Supplementary Table S1.

Five primary color-and-emotion bridge records addressed distinct design functions. Falls-risk feedback combined color with text, icons, and texture [31]. Multimorbidity telemonitoring used a multicolor risk indicator within a complete risk-and-notification workflow [32]. An ePHR study modeled color and icon features as a bundled predictor of usefulness, ease of use, and intention [33]. The A-Kano study embedded bright color within a broader emotional-design package [34]. In the eye-tracking experiment, navigation depth explained more than color, although blue supported more efficient visual exploration [35]. The remaining primary studies examined interface preference [36], accessible color information [37], medication-support themes [38], telepresence contrast [42], and patient-portal legibility for older migrants [44]. Mixed-age evidence concerned graph framing, tuberculosis support, and caregiver self-care [39], [40], [41]. One scoping review remained contextual [43].

Across the resolved matrix, legibility and redundant information were recurring design concerns [31], [37], [42], [44]. Other studies used color for risk status, information hierarchy, navigation, aesthetic preference, or affective tone [32], [34], [35], [36], [38]. These functions imply different mechanisms. Contrast may affect detection, categorical color may affect comprehension, and a palette may influence stated preference. Combining them under one generic color effect would obscure both task and risk.

The causal evidence remained thin. Eight of the ten primary records coded color as not isolated, one as partially isolated, and one as unclear. Nine did not provide an interpretable preference-to-performance test; the remaining study tested the link but did not support it. Five records were directly relevant to safety-critical use, although their outcomes belonged to broader interface or service workflows. None produced a fully isolated color-safety estimate.

V. DISCUSSION

A. A Broad User-Experience Field with a Thin Color Layer

The focal corpus reveals a recognizable bibliometric structure: an international literature concentrated in a small source core and organized around usability, mental health, telehealth, acceptance, and digital service delivery. Its color-specific layer is much smaller than the surrounding affect vocabulary. Resolving candidate records at study level therefore changes the interpretation from topic visibility to documented design evidence. That distinction prevents a prominent metadata term from being treated as participant-level support.

The temporal record helps explain that thinness. Color never formed an independent keyword or citation front; its terms stayed embedded in usability, visualization, accessibility, and experience studies. A variable reported only inside something else accumulates neither shared terminology nor comparable measurement, which is the condition the evidence matrix meets at study level.

The primary studies show why color is better understood through interface function than through a fixed emotional symbol. Accessibility work emphasizes contrast and redundancy [31], [37], [42], [44], while preference and acceptance studies describe stated utility or intention [33], [36], [38]. Risk-status work places color inside a larger decision workflow [32], and the eye-tracking evidence concerns search and trust, with navigation structure explaining more than color [35]. Affective constructs are often invoked without a validated affect measure, and color is rarely manipulated alone. The literature therefore supports testable design hypotheses and reporting standards rather than a claim that one hue is inherently calming, trustworthy, or safe for older users.

B. Implications for Design Evidence and Reporting

Future studies should define the function of color before testing an interface, because each function implies a different design and a different endpoint. A legibility manipulation requires luminance contrast, typography, viewing distance, and task-performance outcomes. A semantic-alert study requires redundant coding, error costs, and comprehension measures. An affective manipulation requires specified color coordinates and a validated emotional outcome.

TABLE IV. PROPOSED RESEARCH RECOMMENDATIONS

Observed reporting issue	Minimum design specification	Proposed validation outcome
Color not quantified	Report color space and coordinates	Replicable visual condition
Contrast absent	Report luminance ratio and text size	Recognition, error, completion time
Affect weakly measured	Use a validated affect measure	Valence, arousal, trust, or stress
Context omitted	State device, illumination, culture	Transferability across settings
Color carries meaning	Add text, shape, icon, or position	Safety and comprehension

Table IV translates the reconciled evidence matrix and its reporting gaps into proposed research recommendations, and reproducibility comes first among them. A named hue such as blue cannot define an experimental condition when displays render different luminance and saturation. A reproducible record specifies the color space and coordinates, foreground and background, contrast ratio, text size, device and screen settings, and ambient illumination; screenshots may supplement those values but cannot replace them. Because older adults meet the same interface under different screens, lighting, and visual capabilities, that context belongs in the record too. WCAG 2.2 offers an accessibility reference [30], yet conformance stops short of empirical testing with intended users, and cultural context deserves equal attention because color meaning is learned and situated.

Outcome alignment matters just as much. Legibility claims need recognition, error, completion-time, or visual-attention evidence, and affective claims need a validated measure of valence, arousal, trust, or anxiety. Adoption, preference, and satisfaction remain useful secondary outcomes, but they cannot stand in for perceptual performance or safety evidence. Critical meaning needs redundancy for the same reason: color can carry hierarchy and rapid scanning, yet a critical state should stay legible through text, shape, icon, position, texture, or sound whenever color perception is uncertain. A rigorous evaluation tests that redundant representation instead of merely mentioning it.

C. Methodological Implications for Bibliometric Studies

Corpus architecture shapes what the analysis can claim. An intersection search would hide the surrounding user-experience literature, whereas a single undifferentiated corpus would let broad affect vocabulary obscure the smaller color-specific layer. Separate denominators preserve context without inflating design evidence, and the same architecture can support other narrowly reported interface variables.

Provider heterogeneity nevertheless limits comparability. All five databases ran the same four conceptual blocks, but field scope and export depth varied, so citation and affiliation coverage differ by provider. Removing the nine PubMed-only records produced only small changes in annual counts and ranking structure. The map is stable to that difference while remaining an assembled corpus rather than a field-neutral union of databases.

D. Toward a Cumulative Evidence Program

A cumulative program begins with descriptive standardization. Authors can name the function of color, state whether it was manipulated or merely present, and report the values that reproduce it, adding such fields to existing checklists without requiring a new theory of color. Retrieval depends on the same discipline: color is often visible in an interface yet invisible in its bibliographic description, so abstracts and keywords should identify color wherever it is manipulated, evaluated, or used to carry meaning, with structured supplementary fields holding the numerical detail.

At a more demanding level, studies need mechanism isolation. When palette, contrast, iconography, wording, and layout change together, no outcome can be attributed to color. A stronger experiment varies one color property while holding semantic content and geometry stable, and factorial designs can test whether lightness, saturation, or hue interacts with task urgency, age-related vision, or affective framing. Manipulation checks should confirm that participants perceived the intended difference before preference or performance is interpreted.

The outcome layers should also be kept apart. Perceptual outcomes include detection, discrimination, reading speed, and visual search; behavioral outcomes include errors, completion, adherence, and recovery from mistakes; affective outcomes include valence, arousal, trust, anxiety, and comfort; clinical outcomes usually require longer follow-up. Separating them keeps a favorable preference rating from being read as evidence of safer or more effective use.

Age heterogeneity deserves its own analysis, since chronological age describes neither contrast sensitivity, color discrimination, and digital literacy nor health status and cultural interpretation. Studies should record the user characteristics theoretically linked to the visual task. Samples rarely support every subgroup, yet planned stratification and transparent uncertainty are preferable to a single assumed older-adult response, and cross-cultural replication matters most where color carries evaluative or risk meaning.

Laboratory evidence also needs an ecological bridge. Controlled screens provide precision rather than the variable illumination, distraction, fatigue, and device settings of daily use. Promising effects should move to field deployment with predefined stopping rules for safety-critical interfaces, and longitudinal work can show whether an initial affective advantage sustains use or fades with novelty.

Such a program also depends on collaboration. The country network shows capacity for international work, and source concentration identifies journals able to normalize better reporting, so design researchers can specify visual variables, clinicians define error costs, and human-factors specialists align outcomes with tasks. The present color matrix offers a starting codebook for that shared language.

For practitioners, the evidence supports a conservative principle that follows the same function-first logic. Endpoints should match the function color serves: search time, route errors, and recall for navigation and hierarchy; detection, comprehension, and error severity for alerts or medication status; and measures separating immediate liking from trust, comfort, or sustained engagement for affective uses. Color can strengthen grouping, salience, identity, and tone, but it should never be the sole carrier of critical information, so design systems should record accessible defaults, redundant cues, tested display conditions, and the circumstances justifying a visual variant.

The bibliometric maps indicate where such standards could diffuse. A small group of sources publishes a large share of the literature and international collaboration is substantial, which makes the source core a plausible site for reporting improvement, though concentration alone says nothing about design quality. The long tail matters equally, since terminology drift is strongest there: a study calling its manipulation a theme, a display setting, or an aesthetic preference leaves no durable keyword, and later retrieval loses the record.

VI. Limitations

The first limitation is unequal search and export depth. As Table I records, Web of Science supplied full records and cited references; Scopus, IEEE Xplore, and ACM were represented largely by result-list metadata; and PubMed matched within title and abstract. Country, institution, citation, and reference networks are consequently shaped more strongly by records with richer fields and by documents found in several databases. The PubMed-only sensitivity analysis showed stable leading sources and keywords, though the five interfaces remain distinct instruments, and the maps describe the assembled corpus rather than each provider's complete native network.

Record linkage introduces a separate qualification. DOI and conservative title evidence were used, and the corrected audit replaces the nonresolving provider DOI in cluster CE-00953 with 10.2196/52442. Records without stable identifiers may still remain separate, and citation values are maximum provider counts captured in July 2026 without field normalization. Author and institution analyses rest on normalized strings rather than ORCID or registry identifiers. The entity audit in Section IV.B shows the cost: address-field fragments split institutions, plausible normalizations can merge distinct ones, and the two parses correlate at rho = 0.759 across the published top thirty. The three leading institutions survive both parses, so rankings deserve confidence at the top and caution below it.

The color matrix has its own constraints. Full text was available for 23 of 41 candidates, so the remaining 17 abstract-level records and one title-level record support less detailed inference. Polysemy and incomplete indexing can alter discovery. The coding protocol improved reproducibility for population and inclusion decisions, yet agreement under one codebook establishes consistency rather than causal validity. The classifier audit adds a second boundary: precision of 0.880 marks the affect stream as deliberately inclusive, and the verb-sense false positives stay identified in S5, so their effect remains traceable. That margin leaves the color layer untouched.

Time and modeling choices add further uncertainty. The 2026 records cover January to July, are reported separately on a monthly basis, and stay outside burst detection. The three thematic periods have unequal duration and volume, because they represent publication phases rather than comparable exposure windows. Change-point testing reproduces the 2023 boundary but places the first anywhere between 2012 and 2018 depending on scale, so the phases function as reporting windows rather than detected regimes. Network appearance likewise depends on thresholds, counting, and layout; the stability analysis separates central nodes, which held in every configuration, from cluster boundaries in the denser semantic maps, which moved with resolution. Records with unverified age or interface-color evidence were excluded rather than imputed, trading sensitivity for interpretability. Future updates should obtain the remaining full texts, preregister the coding protocol, and test whether an external team reproduces the resolved classifications.

VII. Conclusion

This five-database study maps 941 focal records around older adults' health applications, user experience, interaction design, color, and emotion. Output rose sharply, a digital-health source core became visible, and collaboration extended across countries, while usability, mental health, telehealth, acceptance, and service implementation organized the conceptual and intellectual structures. Auditing showed where those descriptions hold: central network nodes and the leading institutions survived changes of parameter and parsing, whereas semantic cluster boundaries and the earlier phase boundary did not. The temporal results indicate growth and differentiation; theoretical and methodological maturity remains an open question.

Color forms a small and uneven evidence layer inside that corpus. The resolved matrix retained ten primary older-adult studies, three mixed-age studies, and one contextual review, and only five color-and-emotion bridge records qualified as primary. Reported functions ranged from legibility and risk communication to navigation, preference, and affective tone, yet most studies changed several interface features at once, so the links from preference to performance and from color to safety remain unresolved. Stronger evidence requires reproducible color values, a defined health task, redundant critical cues, measures matched to the proposed mechanism, and explicit limits on causal attribution.

SUPPLEMENTARY MATERIALS

All supplementary files are openly available through the online repository at https://doi.org/10.5281/zenodo.22798290. Supplementary Table S1 gives the study-level evidence matrix for the 14 retained records. Supplementary Material S2 reproduces the five database expressions with their field tags, limits, execution dates, and result counts, and reports the expanded-vocabulary recall probe. S3 lists the network construction parameters, S4 the stability matrix and change-point tests, S5 the record-level classifier audit, and S6 the entity-identity audit.

DATA AVAILABILITY

The same repository record also holds the search strategies, derived datasets, coding materials, reliability results, and figure source data, and these remain available from the first author on reasonable request. No raw provider export is redistributed, because database licences may restrict redistribution of abstracts and record-level fields.

DECLARATION ON GENERATIVE ARTIFICIAL INTELLIGENCE

The authors used generative artificial intelligence tools to assist with language refinement and with coding the data-processing and analysis scripts used in this study. All such scripts and their outputs are openly available in the repository record cited above. The authors verified every reported result and take full responsibility for the content of this article.

REFERENCES

[1] G. A. Wildenbos, L. Peute, and M. Jaspers, "Aging barriers influencing mobile health usability for older adults: A literature-based framework (MOLD-US)," International Journal of Medical Informatics, vol. 114, pp. 66-75, 2018, doi: 10.1016/j.ijmedinf.2018.03.012.

[2] J. Li, Q. Ma, A. H. Chan, and S. S. Man, "Health monitoring through wearable technologies for older adults: Smart wearables acceptance model," Applied Ergonomics, vol. 75, pp. 162-169, 2019, doi: 10.1016/j.apergo.2018.10.006.

[3] A. Ghorayeb, R. Comber, and R. Gooberman-Hill, "Older adults' perspectives of smart home technology: Are we developing the technology that older people want?," International Journal of Human-Computer Studies, vol. 147, p. 102571, 2021, doi: 10.1016/j.ijhcs.2020.102571.

[4] T. Donker et al., "Smartphones for Smarter Delivery of Mental Health Programs: A Systematic Review," J Med Internet Res, vol. 15, no. 11, p. e247, 2013, doi: 10.2196/jmir.2791.

[5] M. M. Ng, J. Firth, M. Minen, and J. Torous, "User Engagement in Mental Health Apps: A Review of Measurement, Reporting, and Validity," PS, vol. 70, no. 7, pp. 538-544, 2019, doi: 10.1176/appi.ps.201800519.

[6] S. Lehtimaki, J. Martic, B. Wahl, K. T. Foster, and N. Schwalbe, "Evidence on Digital Mental Health Interventions for Adolescents and Young People: Systematic Overview," JMIR Ment Health, vol. 8, no. 4, p. e25847, 2021, doi: 10.2196/25847.

[7] A. J. Elliot and M. A. Maier, "Color Psychology: Effects of Perceiving Color on Psychological Functioning in Humans," Annual Review of Psychology, vol. 65, no. 1, pp. 95-120, 2014, doi: 10.1146/annurev-psych-010213-115035.

[8] P. Valdez, and A. Mehrabian, "Effects of color on emotions," Journal of Experimental Psychology: General, vol. 123, no. 4, pp. 394-409, 1994, doi: 10.1037/0096-3445.123.4.394.

[9] S. E. Palmer, and K. B. Schloss, "An ecological valence theory of human color preference," Proc. Natl. Acad. Sci. U.S.A., vol. 107, no. 19, pp. 8877-8882, 2010, doi: 10.1073/pnas.0906172107.

[10] L. C. Ou, M. R. Luo, A. Woodcock, and A. Wright, "A study of colour emotion and colour preference. Part I: Colour emotions for single colours," Color Research & Application, vol. 29, no. 3, pp. 232-240, 2004, doi: 10.1002/col.20010.

[11] L. C. Ou, M. R. Luo, A. Woodcock, and A. Wright, "A study of colour emotion and colour preference. Part II: Colour emotions for two-colour combinations," Color Research & Application, vol. 29, no. 4, pp. 292-298, 2004, doi: 10.1002/col.20024.

[12] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," PLoS Med, vol. 18, no. 3, p. e1003583, 2021, doi: 10.1371/journal.pmed.1003583.

[13] M. Aria, and C. Cuccurullo, "bibliometrix : An R-tool for comprehensive science mapping analysis," Journal of Informetrics, vol. 11, no. 4, pp. 959-975, 2017, doi: 10.1016/j.joi.2017.08.007.

[14] N. J. v. Eck, and L. Waltman, "Software survey: VOSviewer, a computer program for bibliometric mapping," Scientometrics, vol. 84, no. 2, pp. 523-538, 2010, doi: 10.1007/s11192-009-0146-3.

[15] N. J. v. Eck, L. Waltman, R. Dekker, and J. v. d. Berg, "A comparison of two techniques for bibliometric mapping: Multidimensional scaling and VOS," J. Am. Soc. Inf. Sci., vol. 61, no. 12, pp. 2405-2416, 2010, doi: 10.1002/asi.21421.

[16] A. Perianes-Rodriguez, L. Waltman, and N. J. v. Eck, "Constructing bibliometric networks: A comparison between full and fractional counting," Journal of Informetrics, vol. 10, no. 4, pp. 1178-1195, 2016, doi: 10.1016/j.joi.2016.10.006.

[17] J. Kleinberg, "Bursty and hierarchical structure in streams," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 91-101, 2002, doi: 10.1145/775047.775061.

[18] X. Fan et al., "Utilization of Self-Diagnosis Health Chatbots in Real-World Settings: Case Study," J Med Internet Res, vol. 23, no. 1, p. e19928, 2021, doi: 10.2196/19928.

[19] S. R. Paige, J. L. Krieger, and M. L. Stellefson, "The Influence of eHealth Literacy on Perceived Trust in Online Health Communication Channels and Sources," Journal of Health Communication, vol. 22, no. 1, pp. 53-65, 2017, doi: 10.1080/10810730.2016.1250846.

[20] V. M. Nymberg et al., "'Having to learn this so late in our lives…' Swedish elderly patients' beliefs, experiences, attitudes and expectations of e-health in primary health care," Scandinavian Journal of Primary Health Care, vol. 37, no. 1, pp. 41-52, 2019, doi: 10.1080/02813432.2019.1570612.

[21] S. R. Stoyanov, L. Hides, D. J. Kavanagh, and H. Wilson, "Development and Validation of the User Version of the Mobile Application Rating Scale (uMARS)," JMIR Mhealth Uhealth, vol. 4, no. 2, p. e72, 2016, doi: 10.2196/mhealth.5849.

[22] L. Zhou, J. Bao, I. M. A. Setiawan, A. Saptono, and B. Parmanto, "The mHealth App Usability Questionnaire (MAUQ): Development and Validation Study," JMIR Mhealth Uhealth, vol. 7, no. 4, p. e11500, 2019, doi: 10.2196/11500.

[23] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," Usability Evaluation In Industry, pp. 207-212, 1996, doi: 10.1201/9781498710411-35.

[24] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User Acceptance of Information Technology: Toward A Unified View1," MIS Quarterly, vol. 27, no. 3, pp. 425-478, 2003, doi: 10.2307/30036540.

[25] R. L. Spitzer, K. Kroenke, J. B. W. Williams, and B. Löwe, "A Brief Measure for Assessing Generalized Anxiety Disorder," Arch Intern Med, vol. 166, no. 10, p. 1092, 2006, doi: 10.1001/archinte.166.10.1092.

[26] P. A. Harris et al., "Research electronic data capture (REDCap)-A metadata-driven methodology and workflow process for providing translational research informatics support," Journal of Biomedical Informatics, vol. 42, no. 2, pp. 377-381, 2009, doi: 10.1016/j.jbi.2008.08.010.

[27] J. Torous, J. Nicholas, M. E. Larsen, J. Firth, and H. Christensen, "Clinical review of user engagement with mental health smartphone apps: evidence, theory and improvements," Evid Based Mental Health, vol. 21, no. 3, pp. 116-119, 2018, doi: 10.1136/eb-2018-102891.

[28] J. M. Isautier et al., "People's Experiences and Satisfaction With Telehealth During the COVID-19 Pandemic in Australia: Cross-Sectional Survey Study," J Med Internet Res, vol. 22, no. 12, p. e24531, 2020, doi: 10.2196/24531.

[29] A. Direito, Y. Jiang, R. Whittaker, and R. Maddison, "Apps for IMproving FITness and Increasing Physical Activity Among Young People: The AIMFIT Pragmatic Randomized Controlled Trial," J Med Internet Res, vol. 17, no. 8, p. e210, 2015, doi: 10.2196/jmir.4568.

[30] World Wide Web Consortium, "Web Content Accessibility Guidelines (WCAG) 2.2," W3C Recommendation, Oct. 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/

[31] R. Harte, L. R. Quinlan, L. Glynn, A. Rodríguez-Molinero, et al., "Human-Centered Design Study: Enhancing the Usability of a Mobile Phone App in an Integrated Falls Risk Detection System for Use by Older Adult Users," JMIR mHealth and uHealth, vol. 5, no. 5, p. e71, 2017, doi: 10.2196/mhealth.7046.

[32] M. Chacornac, A. Faoro, J. Texereau, C. Billoet, et al., "Performance of an eHealth (NOMHAD) System Comprising Telemonitoring, Telenotification, and Telecoaching for Patients With Multimorbidity: Proof-of-Concept Study," JMIR Formative Research, vol. 6, no. 3, p. e32205, 2022, doi: 10.2196/32205.

[33] S. Zhang, X. Jia, Z. Zhao, and C. Zhou, "Design features of interfaces and acceptance of ePHR among elderly individuals in China: Insights from an expanded UTAUT framework," DIGITAL HEALTH, vol. 12, 2026, article 20552076261450778, doi: 10.1177/20552076261450778.

[34] X. Liu, B. Qiu, Y.-L. Ling, and K. Su, "Interface Design of Elderly mHealth Apps Based on A-Kano Model and Three-level Emotional Design Theory," Journal of Imaging Science and Technology, vol. 70, no. 1, pp. 1-11, 2026, doi: 10.2352/j.imagingsci.technol.2026.70.1.010409.

[35] B. Wang, J. Li, and Q. Zhu, "Structure Over Color: Eye-Tracking Evidence on Navigation Depth and Elderly Users' Online Trust in Health Apps," IEEE Access, vol. 14, pp. 5026-5036, 2026, doi: 10.1109/ACCESS.2026.3651334.

[36] S. Zhang and J. Song, "An empirical investigation into the preferences of the elderly for user interface design in personal electronic health record systems," Frontiers in Digital Health, vol. 5, 2024, article 1289904, doi: 10.3389/fdgth.2023.1289904.

[37] S. Choi and C. J. Chlebek, "Exploring mHealth design opportunities for blind and visually impaired older users," mHealth, vol. 10, p. 17, 2024, doi: 10.21037/mhealth-23-65.

[38] J. Qin, Z. Wu, H. Zhang, and X. Qi, "Designing age-friendly interfaces and wearable-environment integrations to enhance medication adherence: a discrete choice experiment among older adults," Frontiers in Medicine, vol. 13, 2026, article 1749507, doi: 10.3389/fmed.2026.1749507.

[39] K. Liu, Z. Sun, X. Ren, and D. Tao, "Effects of information framing cues and age on the comprehension of personal health records for self-care behaviors: An eye-tracking study," Journal of the American Medical Informatics Association, vol. 32, no. 7, pp. 1174-1185, 2025, doi: 10.1093/jamia/ocaf085.

[40] V. Balakrishnan, T. Kumaresan, H. Surendra, M. Allsop, et al., "Developing and assessing MyTBCompanion: A tri-lingual integrated video observed therapy app for tuberculosis patient management in Malaysia and Indonesia," PLOS One, vol. 20, no. 4, p. e0320394, 2025, doi: 10.1371/journal.pone.0320394.

[41] M. Liu, T. Jiang, K. Yu, S. Wu, M. Jordan-Marsh, and I. Chi, "Care Me Too, a Mobile App for Engaging Chinese Immigrant Caregivers in Self-Care: Qualitative Usability Study," JMIR Formative Research, vol. 4, no. 12, p. e20325, 2020, doi: 10.2196/20325.

[42] X. Wu and J. M. Beer, "Usability Evaluation of Telepresence Interfaces for Older Adults," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 65, no. 1, pp. 591-595, 2021, doi: 10.1177/1071181321651172.

[43] U. Saxena and N. Sharma, "Elderly User Engagement with Mobile Health Applications: A Qualitative Analysis of User Interface and User Experience for Enhanced Technology Acceptance through a Scoping Review," Journal of Qualitative Research in Health Sciences, vol. 14, article 1346, 2025, doi: 10.34172/jqr.1346.

[44] M. Ghorbanian Zolbin and S. Kujala, "Evaluating the Usability of a Patient Portal: Insights from Older Migrant Adults," Studies in Health Technology and Informatics, vol. 336, pp. 2344-2348, 2026, doi: 10.3233/SHTI260689.

Ablation-Based Climate-Aware Feature Engineering for Rice Productivity Prediction Using Multi-Source Agricultural Data

R. Hadapiningradja Kusumodestoni[1]*, Fatchul Arifin[2], Mutiara Nugraheni[3], Desti Setiyowati[4]
Doctoral Program in Engineering Science, Faculty of Engineering, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia[1, 2, 3]
Faculty of Science and Technology, Universitas Islam Nahdlatul Ulama Jepara, Jepara, Indonesia[1, 4]

***Abstract*—Climate-aware feature engineering is important for rice productivity prediction because rainfall effects may be delayed, accumulated, standardized, or expressed as extreme anomalies. This study evaluates the incremental contribution of rainfall-derived features for rice productivity prediction using multi-source agricultural data from districts/cities in Central Java, Indonesia, during 2010-2025. The analysis used 560 district/city-year observations and compared Random Forest and XGBoost under six sequential ablation scenarios: baseline features without engineered climate variables (S0), rainfall lags (S1), rainfall lags plus three-period moving average (S2), addition of a standardized rainfall anomaly index (S3), addition of rainfall anomaly (S4), and full climate-aware features including extreme dry and wet flags (S5). Rice productivity was defined as average harvested paddy yield per hectare at the district/city-year level, expressed in quintals per hectare, while rice production was excluded from the predictor set to prevent direct target leakage. Rainfall-anomaly parameters were estimated from the training period only, and categorical predictors were treated as nominal variables. The results show that climate-aware features improved XGBoost more strongly than Random Forest. For XGBoost, RMSE decreased from 4.8019 in S0 to 4.3068 in S5, representing a 10.31% reduction. The largest observed improvement occurred after adding rainfall lag features in S1, where RMSE decreased to 4.2121, and R2 increased to 0.3053. However, the differences among several top scenarios were small; therefore, scenario rankings are interpreted descriptively rather than as statistically significant superiority. The study demonstrates that ablation-based feature evaluation is necessary before adopting full climate-aware feature sets in agricultural prediction models.**

***Keywords*—Ablation study; climate-aware feature engineering; multi-source agricultural data; random forest; rainfall lag; rice productivity prediction; standardized rainfall anomaly index; XGBoost**

I. Introduction

Rice productivity prediction is a key analytical task for food-security planning, regional agricultural monitoring, and climate-risk mitigation. Rice remains a strategic staple crop in Indonesia, and changes in productivity can influence food availability, farmer income, market supply, and policy decisions. At the district/city level, productivity does not depend on a single factor. It is shaped by interactions among planted area, harvested area, rainfall, temperature, humidity, sunshine duration, wind speed, soil type, soil pH, drainage, irrigation systems, pest conditions, and temporal climate patterns. Recent climate and agricultural reports also emphasize that food-production systems are increasingly exposed to rainfall variability, drought, floods, and other climate-related disruptions [1], [2]. Therefore, rice productivity prediction requires models that can integrate agricultural, climate, soil-land, agronomic, and temporal information in a valid and interpretable manner.

Machine learning has been widely applied in crop-yield and productivity prediction because it can model non-linear relationships among heterogeneous predictors. Previous studies and reviews have shown that climate indicators, soil properties, management variables, remote-sensing information, and spatial characteristics can improve agricultural prediction when they are properly represented in the feature space [3], [4], [5]. However, strong prediction is not determined only by the algorithm. Feature construction remains essential because agricultural variables often have delayed effects, accumulated effects, and context-dependent interactions. In rice systems, rainfall is especially important because it affects soil moisture, irrigation demand, field preparation, planting timing, nutrient dynamics, pest pressure, and water-management conditions.

Many prediction studies use raw climate variables directly as model inputs. This approach is useful, but it may not fully represent the agronomic meaning of rainfall. Rainfall observed in the current period may not be the only relevant signal because previous rainfall can affect soil-water carry-over, delayed field operations, residual water availability, and crop establishment. Moving averages may represent short-term rainfall tendency, standardized rainfall-anomaly measures can describe dry or wet departures relative to historical conditions, and anomaly variables can capture deviations from expected rainfall patterns. These transformations are consistent with the broader literature on agricultural prediction and climate-risk modelling, which emphasizes that the quality of climate representation can influence model reliability [6], [7], [8].

Tree-based models are suitable for evaluating the value of engineered climate features in tabular agricultural datasets. Random Forest reduces variance through bootstrap aggregation and random feature selection, making it robust for mixed numerical and categorical predictors [9]. XGBoost uses regularized gradient boosting and sequential residual correction, allowing it to exploit informative predictors in structured data [10]. Gradient boosting theory also indicates that staged additive modelling can improve predictive accuracy

*Corresponding author.

when successive learners capture residual patterns left by previous learners [11]. Because these two models are strong tabular learners, they provide a practical basis for testing whether additional climate-aware features produce measurable improvement.

Climate-aware feature engineering should be evaluated empirically rather than assumed to be beneficial. Adding more features can improve model performance when the variables contain complementary information, but it can also introduce redundancy, noise, instability, or overlapping information when extreme-event observations are sparse. This issue is particularly important for rainfall-derived features because rainfall lags, moving averages, anomaly measures, and standardized rainfall indices may partially represent similar rainfall behaviour. Previous studies on crop prediction, smart farming, and climate-aware modelling suggest that data quality, temporal resolution, and feature relevance are often as important as algorithmic complexity [12], [13], [14].

The current research gap is the limited use of ablation-based assessment for climate-aware feature engineering in rice productivity prediction. Many studies report the final set of input variables without showing how each engineered feature group changes model performance. As a result, it remains unclear whether rainfall lags, rainfall moving averages, standardized rainfall anomalies, rainfall anomaly values, and extreme flags contribute equally or whether only some of them provide substantial predictive value. An ablation design can address this problem by adding feature groups step by step and measuring their effect on RMSE, MAE, MAPE, and R^2 [15], [16], [17].

This article is intentionally positioned as a feature-engineering study rather than a model-fusion or explainability article. It is derived from the same broader doctoral research program as companion manuscripts on optimized fusion and explainable AI, but it addresses a different methodological question. The central aim is to evaluate whether climate-aware engineered features improve rice productivity prediction and to identify which feature group contributes the largest observed marginal gain under a fixed temporal evaluation design. The specific objectives are: 1) to construct a leakage-controlled multi-source rice productivity dataset at the district/city-year level; 2) to compare Random Forest and XGBoost across six feature-engineering scenarios; 3) to quantify the incremental contribution of rainfall lags, rainfall moving average, standardized rainfall anomaly, rainfall anomaly, and extreme flags; and 4) to interpret why the full feature set may not always outperform simpler climate-aware scenarios. The expected contribution is practical evidence that rainfall-memory features are valuable, while full climate-aware feature sets should be adopted only after ablation-based validation.

II. Related Work

A. Machine Learning for Crop Productivity Prediction

Machine learning is widely used in crop yield and productivity prediction because it can learn complex relationships from climate, soil, management, and spatial variables. Reviews on crop prediction emphasize that no single algorithm is universally superior; performance depends on data scale, temporal resolution, feature quality, and the match between the model and the data structure [10], [14], [18], In structured agricultural datasets, tree-based models are often competitive because they can handle non-linear relationships and interactions among heterogeneous variables.

Random Forest and XGBoost represent two important families of tree-based learning. Random Forest uses bagging and random feature selection to reduce variance [9], [19], [20]. XGBoost uses regularized gradient boosting to improve prediction by fitting residual errors sequentially [10], [14]. In agricultural modelling, these algorithms are useful for analysing district/city-level tabular data where the number of observations may be limited, and predictors come from different institutional sources.

B. Climate-Aware Feature Engineering

Climate-aware feature engineering transforms raw climate observations into variables that better represent agronomic processes. Rainfall lags can represent temporal water-memory effects, moving averages can capture accumulated rainfall tendencies, rainfall anomalies can express departures from historical means, and standardized rainfall anomalies can describe dry or wet conditions relative to a local reference distribution. These engineered variables are particularly relevant for rice because water availability and drainage conditions directly affect field preparation, vegetative growth, and yield formation.

Previous studies often use climate variables as predictors, but they do not always isolate the marginal contribution of each rainfall-derived feature group. This makes it difficult to determine whether improvement is driven by raw rainfall, delayed rainfall memory, standardized rainfall condition, or extreme-event flags. The present study addresses this limitation by using a sequential ablation design that adds climate-aware feature groups one at a time.

C. Ablation Study, Leakage Control, and Evaluation Stability

Ablation analysis is useful for identifying whether a feature group contributes measurable predictive value. However, a sequential ablation design measures order-dependent marginal contribution rather than independent causal importance. Therefore, ablation results should be interpreted as performance changes under a predefined feature-addition sequence, not as definitive evidence that one feature group is universally more important in all modelling contexts [21], [22], [23].

Data leakage is another important issue in predictive modelling. Leakage can occur when target-related information or future-period information is unintentionally made available during model training [24], [25], [26]. In rice productivity modelling, production is problematic because productivity is mathematically derived from production and harvested area. This study therefore excludes production from the predictor set. The role of harvested area is treated more carefully in this revision because it is part of the productivity denominator and may not always be available at the intended prediction time [27], [28], [29], [30].

Model rankings based on a single temporal split should also be interpreted cautiously. Small differences in RMSE may

reflect sampling variability, scenario order, or the specific train-test period. Repeated seeds, rolling-origin validation, or spatial-temporal validation would provide stronger evidence of stability. Because the present dataset version reports one fixed temporal test period, this manuscript avoids overstating small margins and frames scenario rankings as observed results under the specified split [31], [32].

Table I positions the proposed study within related work. It contrasts common research streams in crop prediction, climate-variable modelling, ablation analysis, and leakage-controlled prediction, then identifies the remaining gap addressed by this manuscript.

TABLE I. POSITION OF THE PROPOSED STUDY WITHIN RELATED WORK

Research stream	Typical contribution	Remaining gap addressed in this study
Tree-based crop prediction	Captures non-linear tabular relationships in agricultural data	Often reports final performance without isolating rainfall-derived feature contributions
Climate-variable modelling	Uses rainfall, temperature, humidity, or drought indicators as predictors	Does not always distinguish raw rainfall, rainfall memory, anomaly, and standardized anomaly effects
Ablation analysis	Evaluates incremental effects of feature groups or model components	Needs cautious interpretation when differences are small and based on one temporal split
Leakage-controlled prediction	Avoids target-related and future-period information in model training	Requires explicit distinction between mathematical leakage and prediction-time availability

Note: The table summarizes how the present study differs from prior crop-prediction and feature-engineering research. The contribution is focused on ablation-based feature assessment rather than developing a new fusion architecture or SHAP interpretation framework.

III. METHODOLOGY

A. Study Area and Dataset

The study used district/city-level agricultural and environmental data from Central Java Province, Indonesia, covering 2010-2025. The research site is in Central Java, which lies approximately between 5.67 degrees S and 8.50 degrees S and 108.50 degrees E-111.50 degrees E, with an approximate provincial centroid at 7.15 degrees S and 110.14 degrees E. The final dataset contained 560 district/city-year observations. Rice productivity was defined as average harvested paddy yield per unit harvested area for each district/city and year, expressed in quintals per hectare (q ha-1).

Agricultural statistics were compiled from BPS-Statistics Indonesia, BPS Province of Central Java, and relevant provincial or district agricultural records. Climate indicators were compiled from meteorological records provided by the Meteorology, Climatology and Geophysical Agency (BMKG) or equivalent station/gridded climate products. The use of BPS and BMKG records is important because these institutions provide official statistical and climate information for Indonesia. Soil-land and agronomic variables were compiled from institutional land-information and agricultural-agency records. All variables were harmonized to a district/city-year unit of analysis before modelling.

Table II presents the main components of the multi-source dataset. The table clarifies how agricultural, climate, soil-land, agronomic, and temporal rainfall features contribute to the ablation experiment before model training.

TABLE II. MAIN COMPONENTS OF THE MULTI-SOURCE DATASET

Data component	Variables or examples	Role in the ablation study
Agricultural data	Planted area, harvested area, productivity	Defines land-use scale and target construction
Climate data	Rainfall, humidity, temperature, sunshine duration, wind speed, extreme rainfall	Represents environmental pressure and climate variability
Soil and land data	Soil type, soil texture, soil pH, soil moisture, drainage type	Represents land suitability and water-management capacity
Agronomic data	Irrigation system and dominant pest type	Represents management and disturbance conditions
Temporal rainfall features	Rain_Lag_1, Rain_Lag_2, Rain_Lag_3, Rain_MA_3, SRAI, Rain_Anomaly and extreme flags	Represents delayed, accumulated, standardized, and abnormal rainfall effects

Note: BPS = Statistics Indonesia; BMKG = Meteorology, Climatology and Geophysical Agency; SRAI = standardized rainfall anomaly index; MA = moving average; Rain_MA_3 = three-period rainfall moving average.

B. Data Source, Preprocessing, and Leakage Control

The unit of analysis was district/city-year. Daily or monthly climate observations were aggregated into annual district/city indicators. When source data had different spatial resolutions, observations were matched using administrative identifiers, year labels, and variable definitions. Quality assurance included duplicate-record checking, unit consistency checking, missing-value inspection, outlier screening, and comparison with official annual trends. The final modelling dataset did not contain missing values.

Production was removed from the input features because rice productivity is mathematically derived from production and harvested area. This step prevents direct target leakage. Harvested area was retained as a land-use scale and prediction-time administrative variable, but its interpretation was revised carefully. Because harvested area is also part of the productivity denominator, the model should not be interpreted as a pure pre-season yield forecast unless harvested area is available before prediction. For stricter pre-season deployment, a sensitivity experiment excluding harvested area should be reported using the same ablation protocol.

TABLE III. PREPROCESSING, LEAKAGE-CONTROL, AND IMPLEMENTATION SAFEGUARDS

Reviewer concern	Revision in methodology	Interpretation safeguard
Target leakage	Production excluded from predictors	Models learn from explanatory variables rather than direct target information
Harvested area availability	Harvested area treated as prediction-time administrative information	For pre-season deployment, a no-harvested-area sensitivity run should be added
Nominal categorical variables	One-hot encoding specified for region, soil, drainage, irrigation, and pest variables	Avoids artificial ordinal relationships from numeric category codes
Rainfall anomaly parameters	Mean and standard deviation estimated from training-period rainfall only	Avoids use of future test-period information

Note: The table summarizes methodological clarifications added in response to concerns about target leakage, categorical encoding, prediction-time availability, and rainfall reference-period estimation.

Categorical variables such as region, soil type, soil texture, soil moisture, drainage type, irrigation system, and dominant pest type were treated as nominal variables. The revised preprocessing protocol specifies one-hot encoding for nominal variables so that tree-based models do not receive arbitrary numerical codes as if they represented ordered categories. Numerical coding, where present in raw files, was used only as an internal label before conversion to indicator variables.

Table III summarizes the methodological safeguards applied during preprocessing and implementation. The table is included to clarify how the study addresses target leakage, harvested-area availability, nominal categorical variables, and training-period rainfall reference parameters.

C. Climate-Aware Feature Engineering

Feature engineering was conducted to represent delayed and accumulated rainfall effects. Rain_Lag_1, Rain_Lag_2, and Rain_Lag_3 represented rainfall from previous periods. Rain_MA_3 represented a three-period moving average of rainfall. Rain_Anomaly represented the deviation of rainfall from the training-period mean. The previously labelled SPI-style feature is reported in this revision as a standardized rainfall anomaly index (SRAI), because it is calculated as a rainfall z-score and does not implement the full conventional SPI procedure involving a specified accumulation scale, distribution fitting, cumulative probability transformation, and standard-normal transformation.

Rainfall anomaly and SRAI were calculated using reference parameters estimated only from the training period. Specifically, the district-specific training-period mean rainfall and standard deviation were used to transform both training and test observations. This prevents the 2023-2025 test-period rainfall distribution from influencing feature construction. The extreme dry and wet flags were then derived from the same training-period reference logic, so their thresholds remained fixed when applied to future test observations.

D. Ablation Scenarios for Climate-Aware Feature Engineering

Six feature-engineering scenarios were constructed to isolate the incremental contribution of climate-aware variables. Scenario S0 used base features without engineered climate-aware variables. Scenario S1 added rainfall lag variables. Scenario S2 added rainfall moving average. Scenario S3 added SRAI. Scenario S4 added rainfall anomaly. Scenario S5 added extreme dry and wet flags, representing the full climate-aware feature set.

The ablation design is sequential. Therefore, the reported contribution of each feature group is conditional on its position in the sequence. The design should not be interpreted as independent feature importance or causal attribution. Instead, it measures how model performance changes when each feature group is added after the preceding groups under an identical temporal split and fixed model configuration.

Table IV defines the six ablation scenarios used in the experiment. The table shows the sequential addition of rainfall lags, rainfall moving average, standardized rainfall anomaly index, rainfall anomaly, and extreme-condition flags.

Fig. 1 illustrates the overall ablation workflow for climate-aware rainfall features. The diagram shows how the experiment starts from the base feature set and gradually adds engineered rainfall variables before evaluating Random Forest and XGBoost under the same independent temporal test period.

TABLE IV. ABLATION SCENARIOS FOR CLIMATE-AWARE FEATURE ENGINEERING

Scenario	Feature set	Purpose
S0	Base features without climate-aware engineered variables	Baseline comparison
S1	S0 + Rain_Lag_1, Rain_Lag_2 and Rain_Lag_3	Evaluate rainfall-memory effects
S2	S1 + Rain_MA_3	Evaluate smoothed rainfall tendency
S3	S2 + SRAI	Evaluate standardized rainfall departure using a training-period reference
S4	S3 + Rain_Anomaly	Evaluate deviation from training-period rainfall mean
S5	S4 + Extreme_Dry_Flag and Extreme_Wet_Flag	Evaluate binary extreme-condition signals

Note: S0-S5 = sequential ablation scenarios; SRAI = standardized rainfall anomaly index; Rain_MA_3 = three-period rainfall moving average.

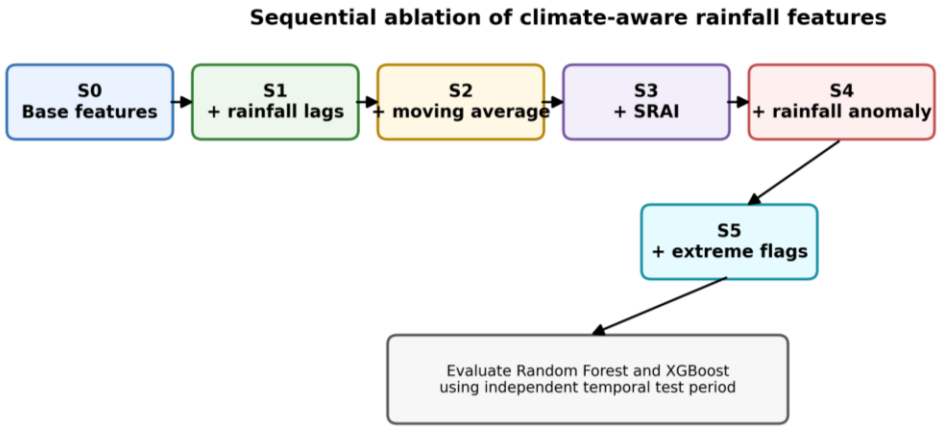


Fig. 1. Ablation design for climate-aware feature engineering.

E. Models and Hyperparameter Settings

Random Forest and XGBoost were selected because both models are strong tabular learners and are suitable for assessing the contribution of additional feature groups. Random Forest was used as a variance-reduction learner for non-linear tabular patterns [9], [33]. XGBoost was used as a regularized boosting learner for structured-data prediction [11], [34]. Each model was trained and evaluated under all six scenarios using the same temporal split so that differences in performance could be attributed primarily to feature-set changes rather than changes in validation design.

Data from 2010-2022 were used for model training, and data from 2023-2025 were used for independent temporal testing. Hyperparameters were kept stable across ablation scenarios to focus the comparison on feature contribution rather than repeated model re-tuning. This design supports a fair ablation comparison, but it does not replace repeated temporal validation or statistical significance testing across multiple splits [19], [20], [35], [36].

Table V reports the model configuration used for all ablation scenarios. The table confirms that Random Forest, XGBoost, temporal validation, and categorical encoding settings were kept consistent so that performance changes mainly reflect feature-set changes.

TABLE V. MODEL CONFIGURATION FOR THE ABLATION EXPERIMENT

Model or process	Configuration used for reporting
Random Forest	n_estimators = 500; max_depth = None; min_samples_split = 2; min_samples_leaf = 1; max_features = sqrt; bootstrap = True; random_state = 42. Purpose: robust tabular baseline and comparison learner.
XGBoost	objective = reg:squarederror; n_estimators = 300; max_depth = 3; learning_rate = 0.05; subsample = 0.80; colsample_bytree = 0.80; reg_lambda = 1.0; random_state = 42. Purpose: regularized boosting learner for structured agricultural data.
Validation design	Training period = 2010-2022; test period = 2023-2025; identical split applied across S0-S5. Purpose: temporal evaluation without future-information leakage.
Categorical encoding	Nominal predictors converted into one-hot indicators before model fitting. Purpose: avoid artificial ordinal relations from numeric category codes.

Note: XGBoost = Extreme Gradient Boosting; sqrt = square root; reg:squarederror = regression with squared-error objective.

F. Mathematical Formulation and Evaluation Metrics

To strengthen reproducibility, this subsection presents the equations used for target construction, climate-aware feature engineering, and performance evaluation. Let d denote district/city, t denote year, i denote a district/city-year observation, y_i denote observed productivity, and yhat_i denote predicted productivity. Eq. (1)-(10) are cited sequentially according to their function.

Productivity is calculated as harvested paddy yield per hectare at the district/city-year level as shown in Eq. (1). Production was excluded from the input features because it is a direct component of the target construction.

$$Productivity_{d,t} = (Paddy\ production_{d,t} / Harvested\ area_{d,t}) \times 10 \quad (1)$$

$$Rain\_Lag_{k,d,t} = Rainfall_{d,t-k}, \text{ where } k = 1, 2, 3 \quad (2)$$

$$Rain\_MA\_3_{d,t} = (Rainfall_{d,t} + Rainfall_{d,t-1} + Rainfall_{d,t-2}) / 3 \quad (3)$$

$$Rain\_Anomaly_{d,t} = Rainfall_{d,t} - mu_{d,train} \quad (4)$$

$$SRAI_{d,t} = (Rainfall_{d,t} - mu_{d,train}) / sigma_{d,train} \quad (5)$$

$$RMSE = \sqrt{[(1/n)\ sum_{i=1}^{n}\ (y_i - yhat_i)^2]} \quad (6)$$

$$MAE = (1/n)\ sum_{i=1}^{n}\ |y_i - yhat_i| \quad (7)$$

$$MAPE = (100/n)\ sum_{i=1}^{n}\ |(y_i - yhat_i) / y_i| \quad (8)$$

$$R2 = 1 - [sum_{i=1}^{n}\ (y_i - yhat_i)^2 / sum_{i=1}^{n}\ (y_i - ybar)^2] \quad (9)$$

$$RMSE\ reduction\ (\%) = [(RMSE\_S0 - RMSE\_Sj) / RMSE\_S0] \times 100 \quad (10)$$

Eq. (2)-(5) define the engineered rainfall features. Eq. (6)-(9) define RMSE, MAE, MAPE, and R^2. Eq. (10) calculates the percentage reduction in RMSE relative to the baseline scenario S0. The terms mu_{d,train} and sigma_{d,train} represent district-specific rainfall mean and standard deviation estimated exclusively from the training period.

IV. Results and Discussion

A. Overall Ablation Results

The ablation results show that climate-aware feature engineering has different effects on Random Forest and XGBoost. For Random Forest, RMSE decreased from 4.4466 in S0 to 4.3732 in S5, equivalent to a 1.65% reduction. The lowest observed Random Forest RMSE occurred in S4 after adding rainfall anomaly, with RMSE of 4.3578. This suggests that Random Forest benefited from climate-aware variables, but the gain was modest and the performance curve remained relatively flat across scenarios.

For XGBoost, the effect was stronger. The base scenario produced an RMSE of 4.8019 and a a MAPE of 6.0520%. After adding rainfall lag features in S1, RMSE decreased to 4.2121, MAPE decreased to 5.3318%, and R^2 increased to 0.3053. This was the largest observed marginal gain across all scenarios. The lowest observed XGBoost RMSE occurred in S1, whereas the lowest observed XGBoost MAPE occurred in S3 after adding SRAI. These results indicate that rainfall-memory features are the most important engineered climate feature group in the present temporal split.

Table VI reports the complete ablation results for Random Forest and XGBoost. The table allows direct comparison of RMSE, MAE, MAPE, and R2 across S0-S5, while also showing that the full climate-aware feature set is not automatically the best observed scenario.

Fig. 2 visualizes RMSE across the six climate-aware feature-engineering scenarios. The figure highlights the sharp XGBoost improvement after adding rainfall lags and the relatively flat Random Forest curve across subsequent feature additions.

Fig. 3 presents MAPE across the same ablation scenarios. The figure shows that XGBoost obtains its lowest observed relative error after SRAI is added, while Random Forest remains comparatively stable across climate-aware feature sets.

TABLE VI. Ablation Results Across Climate-Aware Feature-Engineering Scenarios

Scenario	Model	Features	RMSE	MAE	MAPE (%)	R2
S0 Base features	Random Forest	17	4.4466	3.3885	5.9629	0.2259
S0 Base features	XGBoost	17	4.8019	3.5047	6.0520	0.0972
S1 Base + rain lags	Random Forest	20	4.3690	3.3267	5.8614	0.2526
S1 Base + rain lags	XGBoost	20	4.2121	3.0644	5.3318	0.3053
S2 S1 + Rain_MA_3	Random Forest	21	4.3699	3.3411	5.8996	0.2523
S2 S1 + Rain_MA_3	XGBoost	21	4.2942	3.1249	5.4375	0.2780
S3 S2 + SRAI	Random Forest	22	4.3692	3.3527	5.9190	0.2526
S3 S2 + SRAI	XGBoost	22	4.2310	3.0563	5.3250	0.2991
S4 S3 + anomaly	Random Forest	23	4.3578	3.3399	5.8981	0.2565
S4 S3 + anomaly	XGBoost	23	4.4175	3.1873	5.5101	0.2359
S5 Full climate-aware	Random Forest	25	4.3732	3.3530	5.9226	0.2512
S5 Full climate-aware	XGBoost	25	4.3068	3.2007	5.5767	0.2737

Note: RMSE = root mean squared error; MAE = mean absolute error; MAPE = mean absolute percentage error; R2 = coefficient of determination. Values are reported to four decimal places to avoid overstating precision from a single temporal split.

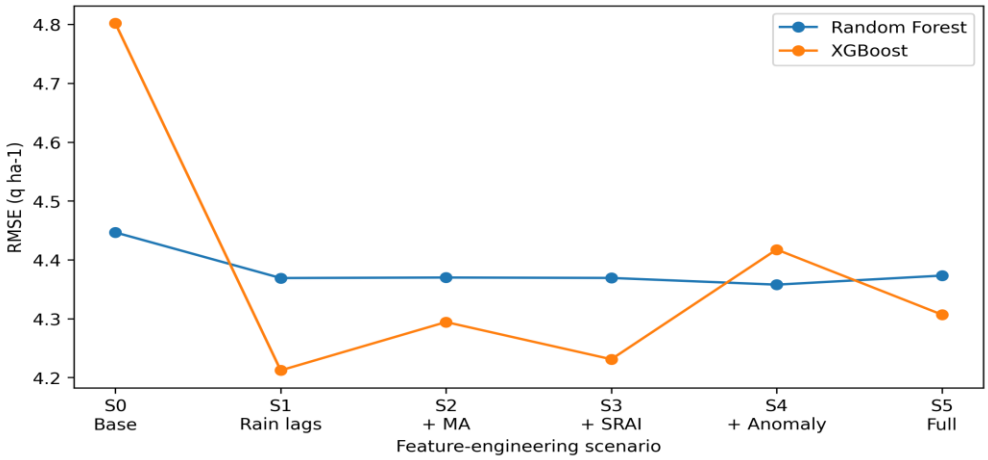


Fig. 2. RMSE across climate-aware feature-engineering scenarios.

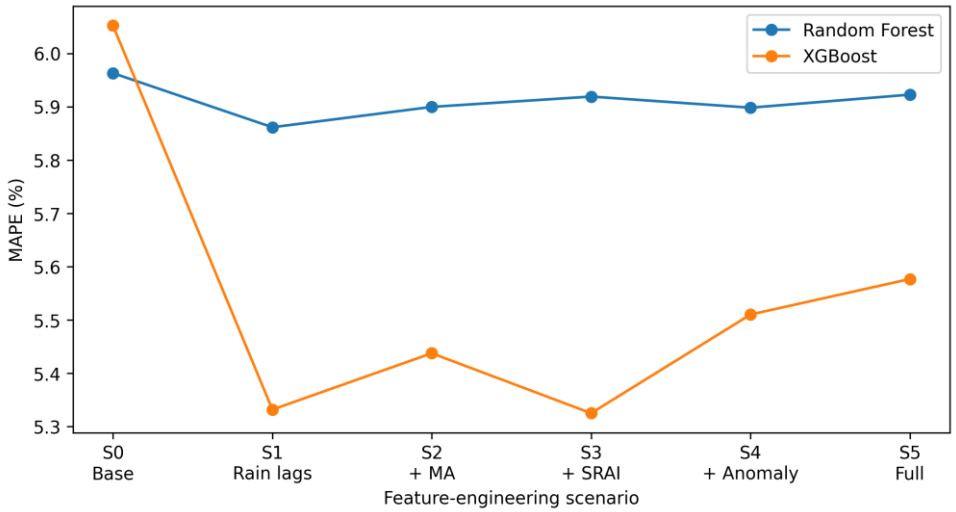


Fig. 3. MAPE across climate-aware feature-engineering scenarios.

B. *Improvement from Baseline to Full Climate-Aware Features*

Table VII compares the baseline scenario with the full climate-aware scenario. XGBoost achieved an RMSE reduction of 0.4951, corresponding to a 10.31% improvement. Random Forest achieved a smaller RMSE reduction of 0.0734, corresponding to a 1.65% improvement. This difference indicates that XGBoost is more responsive to engineered climate features than Random Forest in the present dataset.

The improvement should be interpreted carefully. Although S5 improved substantially over S0 for XGBoost, S5 did not outperform S1 in RMSE or S3 in MAPE. Therefore, the full feature set provides evidence that climate-aware information is useful, but it does not prove that every engineered feature group contributes independent predictive value.

Table VII compares the baseline feature set with the full climate-aware feature set. This table quantifies the RMSE reduction from S0 to S5 and demonstrates that XGBoost is more responsive to engineered rainfall features than Random Forest in the present dataset.

Fig. 4 shows incremental RMSE reduction relative to the S0 baseline. The figure helps explain that the largest observed gain occurs immediately after adding rainfall lag variables, whereas later additions produce smaller or less stable improvements.

TABLE VII. RMSE REDUCTION FROM BASELINE TO FULL CLIMATE-AWARE FEATURE SET

Model	RMSE S0	RMSE S5	RMSE reduction	RMSE reduction (%)
Random Forest	4.4466	4.3732	0.0734	1.65
XGBoost	4.8019	4.3068	0.4951	10.31

Note: S0 = base features without climate-aware engineered variables; S5 = full climate-aware feature set; RMSE reduction was calculated using Eq. (10).

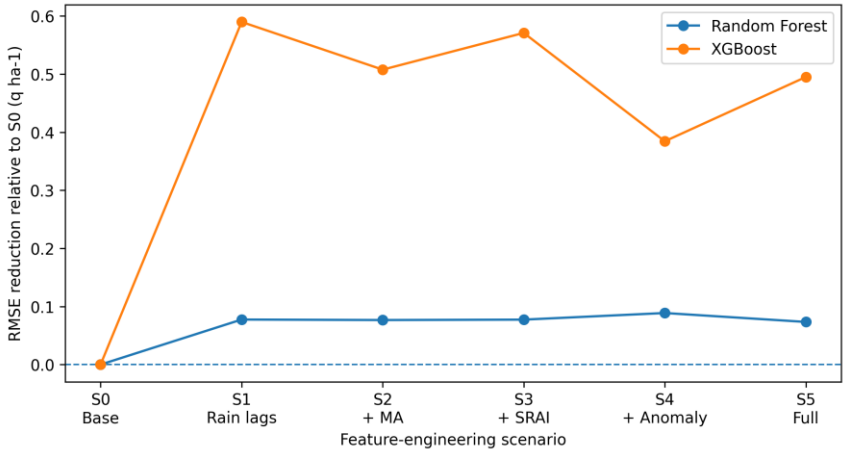


Fig. 4. Incremental RMSE reduction relative to the baseline scenario.

C. Best-Observed Scenarios and Stability Caution

The best scenario depends on both the model and the metric. For Random Forest, S4 produced the lowest observed RMSE, suggesting that rainfall anomaly provided a small additional benefit. For XGBoost, S1 produced the lowest observed RMSE, indicating that rainfall lag variables captured the strongest predictive signal. For MAPE, the best observed XGBoost result occurred in S3, suggesting that SRAI improved relative error even when the RMSE difference was small.

However, several observed differences are small. For Random Forest, S1, S2, S3, and S5 are within approximately 0.0154 RMSE of the lowest observed S4 value. For XGBoost, S3 is within approximately 0.0189 RMSE of S1. These gaps should not be interpreted as statistically significant evidence because the present experiment uses one fixed temporal split. The results therefore identify promising feature sets rather than definitive universal rankings.

Table VIII summarizes the best-observed scenarios by metric under the fixed temporal split. The table is interpreted cautiously because the scenario differences were not validated through repeated seeds, rolling-origin evaluation, or statistical significance testing.

TABLE VIII. BEST-OBSERVED SCENARIOS BY METRIC UNDER THE FIXED TEMPORAL SPLIT

Criterion	Model	Scenario	RMSE	MAE	MAPE (%)	R2	Interpretation
Lowest observed RMSE	Random Forest	S4 S3 + anomaly	4.3578	3.3399	5.8981	0.2565	Very small margin over other RF climate-aware scenarios
Lowest observed RMSE	XGBoost	S1 Base + rain lags	4.2121	3.0644	5.3318	0.3053	Strongest observed rainfall-memory gain
Lowest observed MAPE	Random Forest	S1 Base + rain lags	4.3690	3.3267	5.8614	0.2526	Observed relative-error advantage
Lowest observed MAPE	XGBoost	S3 S2 + SRAI	4.2310	3.0563	5.3250	0.2991	Small relative-error improvement over S1

Note: The term best-observed is used deliberately because repeated seeds, rolling-origin validation, and statistical significance tests were not available in the present dataset version.

D. Explanatory Power and Prediction-Time Availability

The best R^2 value across all scenarios was 0.3053 for XGBoost under S1. This means that the model explained only part of the variance in rice productivity, while most variation remained outside the available predictors. The result should not be interpreted as model failure; rather, it indicates that district/city-level rice productivity is affected by factors that were not fully represented in the dataset, such as rice variety, fertilizer dose and timing, irrigation volume, mechanization, labour availability, planting calendar, pest intensity, field-level management, and satellite-derived vegetation indices.

This explanatory ceiling also changes the interpretation of RMSE reduction. A 10.31% reduction from S0 to S5 for XGBoost indicates that engineered rainfall information improves prediction error relative to the baseline, but it does not imply that the model captures all determinants of rice productivity. Therefore, the practical contribution of this study is a feature-engineering assessment, not a claim of complete productivity explanation.

Harvested area also requires careful interpretation. Because productivity is calculated from production divided by harvested area, harvested area is mathematically related to the target construction. In this manuscript, production is excluded to avoid direct target leakage, while harvested area is retained only when it is available at prediction time as an administrative land-use variable. For strict ex-ante forecasting before harvested area is known, the same S0-S5 ablation should be repeated without harvested area. Until that sensitivity analysis is added, conclusions should be interpreted as conditional on the included prediction-time variables.

Table IX provides interpretation safeguards for the main reviewer concerns. It summarizes how the manuscript treats the R2 ceiling, small RMSE margins, harvested-area availability, and sparse extreme flags to avoid overclaiming from a single temporal split.

TABLE IX. INTERPRETATION SAFEGUARDS FOR REVIEWER CONCERNS

Issue	Current treatment in this manuscript	Recommended interpretation
R2 ceiling	Best observed R2 = 0.3053 for XGBoost S1	Model explains part of district/city-year productivity variation; omitted variables remain important
Small RMSE margins	Several climate-aware scenarios are separated by less than 0.10 RMSE	Use descriptive rather than statistically definitive ranking
Harvested area	Retained as prediction-time administrative land-use information	Repeat ablation without harvested area for strict pre-season forecasting
Extreme flags	Included only in S5 after anomaly features	Interpret cautiously because extreme observations are sparse

Note: These safeguards prevent overclaiming from a single temporal split and clarify the distinction between target-leakage prevention and prediction-time variable availability.

E. Interpretation of Feature-Engineering Contribution

The ablation results indicate that not all engineered climate features contribute equally. Rainfall lag features provided the most consistent and substantial improvement, particularly for XGBoost. This is agronomically meaningful because rice productivity can be influenced by previous rainfall conditions through soil moisture, irrigation availability, field preparation, planting timing, and pest dynamics. The strong effect of S1 suggests that rainfall memory is a core climate signal for district/city-year productivity prediction.

Adding Rain_MA_3 did not improve XGBoost beyond S1, indicating that rainfall lag variables already captured much of the useful short-term rainfall information. Adding SRAI in S3 slightly improved MAPE compared with S1, suggesting that standardized rainfall status is useful for relative error control. However, adding rainfall anomaly and extreme flags did not further improve XGBoost performance and increased error relative to S1 and S3. This may be caused by redundancy among rainfall-derived variables, the small number of extreme observations, or additional feature noise.

For Random Forest, the performance curve was relatively flat across scenarios. This suggests that Random Forest was less sensitive to incremental climate-aware features. XGBoost, by contrast, showed a sharper response to rainfall lags and SRAI, which is consistent with the ability of boosting models to exploit informative residual patterns from engineered predictors. These findings support a practical recommendation: climate-aware feature engineering should be adopted selectively, and rainfall-memory variables should be prioritized before more complex climate indices are added.

F. Discussion and Relationship to the Broader Research Program

The main finding of this article is that climate-aware feature engineering improves rice productivity prediction, but the improvement is not monotonic. The strongest observed gain comes from adding rainfall lag features, while the full feature set does not necessarily produce the best performance. This finding challenges the assumption that adding more engineered climate variables always improves model accuracy. In practical modelling workflows, additional variables should be evaluated not only by theoretical relevance but also by their empirical contribution under temporal testing.

This study is complementary to broader work on rice productivity prediction conducted within the same doctoral research program, but it has a distinct scope. The optimized-fusion manuscript focuses on combining RF, XGBoost, and LSTM outputs. The explainability manuscript focuses on interpreting model drivers using SHAP. The present manuscript focuses on upstream climate-aware feature engineering and asks which rainfall-derived feature groups improve tabular models before any final fusion or detailed explanation stage. This distinction reduces overlap and clarifies the manuscript contribution.

Several limitations should be noted. First, the dataset uses annual district/city-level observations, which may not fully represent intra-seasonal rainfall timing, rice phenology, planting-calendar variation, or field-level management. Second, the number of extreme observations is limited, so extreme flags may not provide stable incremental information. Third, the study uses a single temporal test split; repeated seeds, rolling-origin evaluation, or spatiotemporal cross-validation would provide stronger evidence for scenario stability. Fourth, the no-harvested-area sensitivity analysis should be added before claiming strict pre-season applicability. Fifth, model hyperparameters were kept stable across scenarios to focus on feature contribution; alternative tuning strategies may produce different results.

V. CONCLUSION

This study evaluated climate-aware feature engineering for rice productivity prediction using a sequential ablation design and multi-source agricultural data from Central Java, Indonesia. Six scenarios were tested, ranging from base features to a full climate-aware feature set containing rainfall lags, rainfall moving average, standardized rainfall anomaly index, rainfall anomaly, and extreme dry/wet flags. Random Forest and XGBoost were used as tabular learners to isolate the contribution of feature groups under the same temporal testing design.

The results show that XGBoost benefited substantially from climate-aware feature engineering. Full climate-aware features reduced XGBoost RMSE by 10.31% compared with the base feature set. However, the lowest observed XGBoost RMSE was achieved by S1, which added rainfall lag variables, not by the full S5 feature set. This indicates that rainfall lag features provide the largest observed marginal contribution, while adding all engineered features can introduce redundancy, sparse-signal instability, or noise. Random Forest showed smaller improvement, with only 1.65% RMSE reduction from S0 to S5.

The study also emphasizes methodological caution. The best observed R^2 was 0.3053, indicating that much productivity variation remains unexplained by the available predictors. Several differences among scenarios were small and should not be interpreted as statistically significant without repeated evaluation. Harvested area was retained only as a prediction-time administrative variable, and a sensitivity analysis without harvested area is recommended for strict pre-season forecasting. Future research should test the same feature-engineering scheme using monthly or seasonal rainfall data, alternative standardized-rainfall anomaly time scales, spatial-temporal validation, remote-sensing indicators, richer agronomic variables, and repeated-split statistical testing.

ACKNOWLEDGMENT

The authors thank Universitas Negeri Yogyakarta and the institutions providing agricultural and climate data used in this research. The authors also acknowledge the academic supervision and methodological feedback received during the dissertation development process.

DECLARATIONS

AUTHORS' CONTRIBUTIONS

Hadapiningradja Kusumodestoni Contributed to Conceptualization, Data Curation, Methodology, Analysis, Visualization, and Original Draft Preparation. Fatchul Arifin Contributed to Supervision, Methodology Validation, and Manuscript Review. Mutiara Nugraheni Contributed to Supervision, Interpretation, and Manuscript Review. Desti Setiyowati Contributed to Data Review and Manuscript Preparation Support. All Authors Read and Approved the Final Manuscript.

CONFLICT OF INTERESTS

The authors declare no conflict of interest.

DATA AVAILABILITY

The processed dataset, feature-engineering scenario table, model-output files, and Python scripts used in this study are available from the corresponding author upon reasonable request. Public release of the complete dataset may depend on data-sharing permissions from the original data providers.

Declaration of Artificial Intelligence Assistance: The authors used AI-assisted language support to improve grammar, formatting consistency, and editorial clarity during revision. No AI tool was used to create the research data, conduct the statistical analysis, or make conceptual decisions. All data analysis, interpretation, and final manuscript responsibility remain with the authors.

Ethics Approval: This study used secondary agricultural, climate, soil, and agronomic data and did not involve human participants, patient information, or animal experiments. Therefore, human or animal ethics approval was not applicable.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] ADDIN Mendeley Bibliography CSL_BIBLIOGRAPHY [1] World Meteorological Organization, State of the Global Climate 2024. Geneva, Switzerland: World Meteorological Organization, 2025, WMO No. 1368. [Online]. Available: https://wmo.int/sites/default/files/2025 03/WMO-1368-2024_en.pdf.

[2] BPS-Statistics Indonesia, Statistical Yearbook of Indonesia 2025. Jakarta, Indonesia: BPS-Statistics Indonesia, 2025, Catalogue No. 1101001, Publication No. 03200.25004. [Online]. Available: https://www.bps.go.id/en/publication/2025/02/28/8cfe1a589ad3693396d 3db9f/statistical-yearbook-of-indonesia-2025.html.

[3] M. Rashid, B. S. Bari, Y. Yusup, M. A. Kamaruddin, and N. Khan, "A Comprehensive Review of Crop Yield Prediction Using Machine Learning Approaches With Special Emphasis on Palm Oil Yield Prediction," vol. 9, 2021, doi: 10.1109/ACCESS.2021.3075159.

[4] U. Shafi *et al.*, "Tackling Food Insecurity Using Remote Sensing and Machine Learning-Based Crop Yield Prediction," *IEEE Access*, vol. 11, no. October, pp. 108640–108657, 2023, doi: 10.1109/ACCESS.2023.3321020.

[5] J. Lu *et al.*, "Deep Learning for Multi-Source Data-Driven Crop Yield Prediction in Northeast China," 2024.

[6] C. Lin, Z. Yan, and S. Miao, "Identifying Nonlinear Thresholds and Interaction Dominance of Meteorological Drivers on Rice Yield : A SHAP-Based Approach," pp. 1–30, 2026.

[7] K. G. Liakos, P. Busato, D. Moshou, and S. Pearson, "Machine Learning in Agriculture : A Review," no. Ml, pp. 1–29, doi: 10.3390/s18082674.

[8] T. Van Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning : A systematic literature review," *Comput. Electron. Agric.*, vol. 177, no. January, p. 105709, 2020, doi: 10.1016/j.compag.2020.105709.

[9] L. E. O. Breiman, "Random Forests," pp. 5–32, 2001.

[10] T. Chen and C. Guestrin, "XGBoost : A Scalable Tree Boosting System," pp. 785–794, 2016.

[11] V. Nguyen, C. Le, M. T. Bui, T. Nguyen, and T. Le, "Ecological Informatics Multi-modal representation for UAV-based rice yield prediction : Understanding temporal redundancy and feature sufficiency," *Ecol. Inform.*, vol. 98, no. July, p. 103957, 2026, doi: 10.1016/j.ecoinf.2026.103957.

[12] H. Demirhan, "Information Processing in Agriculture: A deep learning framework for prediction of crop yield in Australia under the impact of climate change," *Inf. Process. Agric.*, vol. 12, no. 1, pp. 125–138, 2025, doi: 10.1016/j.inpa.2024.04.004.

[13] S. Wolfert, L. Ge, C. Verdouw, and M. Bogaardt, "Big Data in Smart Farming – A review," *Agric. Syst.*, vol. 153, pp. 69–80, 2017, doi: 10.1016/j.agsy.2017.01.023.

[14] M. Waqas, S. Ahmad, A. Wangwongchai, and P. Dechpichai, "Quantification and prediction of the impact of ENSO on rainfed rice yields in Thailand," *Environ. Challenges*, vol. 19, no. January, p. 101123, 2025, doi: 10.1016/j.envc.2025.101123.

[15] M. A. Jabed and M. A. Azmi Murad, "Crop yield prediction in agriculture: A comprehensive review of machine learning and deep learning approaches, with insights for future research and sustainability," *Heliyon*, vol. 10, no. 24, p. e40836, 2024, doi: 10.1016/j.heliyon.2024.e40836.

[16] G. Gupta and S. Kumar, "Applications of AI in precision agriculture," *Discov. Agric.*, no. November 2024, 2025, doi: 10.1007/s44279-025-00220-9.

[17] H. Li, J. Gao, Y. Guo, and X. G. Yuan, "Application of XGBoost model and multi-source data for winter wheat yield prediction in Henan Province of China," no. March, pp. 29–47, 2025.

[18] R. Hadapiningradja Kusumodestoni, F. Arifin and M. Nugraheni, "Optimized Weighted Fusion for Rice Productivity Prediction Using Multi-source Agricultural Data," vol. 7, no. July, pp. 1530–1546, 2026, doi: 10.47857/irjms.2026.v07i03.012859.

[19] R. Hadapiningradja Kusumodestoni, F. Arifin and M. Nugraheni, "Comparative analysis of random forest and XGBoost for rice yield estimation," vol. 020002, no. November 2025, 2026.

[20] Y. Prediction, "A Temporal – Geospatial Deep Learning Framework for Crop Yield Prediction," 2024.

[21] J. Brinkhoff, A. Clarke, B. W. Dunn, and M. Groat, "Agricultural and Forest Meteorology Analysis and forecasting of Australian rice yield using phenology-based aggregation of satellite and weather data," vol. 353, no. May, 2024.

[22] Y. Liu, C. Ao, W. Zeng, Z. Li, Y. Zhang, and J. Ishchanov, "Assimilation of remote sensing data and crop model simulations for improved rice yield prediction under controlled irrigation and extreme precipitation in Northeast China," vol. 333, no. June, 2026.

[23] T. Balasubramaniam, W. A. Mohotti, K. Sabir, and R. Nayak, "Ecological Informatics Feature engineering on climate data with machine learning to understand time-lagging effects in pasture yield prediction," *Ecol. Inform.*, vol. 86, no. September 2024, p. 103011, 2025, doi: 10.1016/j.ecoinf.2025.103011.

[24] S. Kaufman, S. Rosset, C. Perlich, and O. R. I. Stitelman, "Leakage in Data Mining : Formulation , Detection , and Avoidance," vol. 6, no. 4, pp. 1–21, 2012, doi: 10.1145/2382577.2382579.

[25] P. Probst, "Tunability : Importance of Hyperparameters of Machine Learning Algorithms," vol. 20, pp. 1–32, 2019.

[26] Y. Li, H. Zeng, M. Zhang, B. Wu, Y. Zhao, and X. Yao, "International Journal of Applied Earth Observation and Geoinformation A county-level soybean yield prediction framework coupled with XGBoost and multidimensional feature engineering," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 118, no. August 2022, p. 103269, 2023, doi: 10.1016/j.jag.2023.103269.

[27] L. Fan, S. Fang, J. Fan, Y. Wang, L. Zhan, and Y. He, "Rice Yield Estimation Using Machine Learning and Feature Selection in Hilly and Mountainous Chongqing , China," 2024.

[28] M. N. Id, A. Hossain, C. Id, F. Arefeen, and M. Id, "A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh," pp. 1–15, 2023, doi: 10.1371/journal.pone.0283452.

[29] J. W. Choi *et al.*, "Recent Trends in Machine Learning , Deep Learning , Ensemble Learning , and Explainable Artificial Intelligence Techniques for Evaluating Crop Yields Under Abnormal Climate Conditions," pp. 1–26, 2025.

[30] R. Zeng, J. Li, Z. Li, and Q. Zhang, "Crossformer-Based Model for Predicting and Interpreting Crop Yield Variations Under Diverse Climatic and Agricultural Conditions," pp. 1–26, 2025.

[31] P. Probst, "Hyperparameters , Tuning and Meta-Learning for Random Forest and Other Machine Learning Algorithms," 2019.

[32] M. Waqas, S. Ahmad, A. Wangwongchai, and P. Dechpichai, "Quantification and prediction of the impact of ENSO on rainfed rice yields in Thailand," *Environ. Challenges*, vol. 19, no. February, p. 101123, 2025, doi: 10.1016/j.envc.2025.101123.

[33] A. Amarasinghe, I. Sangarasekara, N. De Silva, M. Ariyaratne, and R. Amarasinghe, "Advancing food sustainability : a case study on improving rice yield prediction in Sri Lanka using weather - based , feature - engineered machine learning models," *Discov. Appl. Sci.*, 2024, doi: 10.1007/s42452-024-06300-7.

[34] S. Sah, D. Haldar, R. N. Singh, B. Das, and A. S. Nain, "Rice yield prediction through integration of biophysical parameters with SAR and optical remote sensing data using machine learning models," pp. 1–17, 2024.

[35] H. Fang, S. Liang, Y. Chen, H. Ma, W. Li, and T. He, "Science of Remote Sensing A comprehensive review of rice mapping from satellite data : Algorithms , product characteristics and consistency assessment," *Sci. Remote Sens.*, vol. 10, no. July, p. 100172, 2024, doi: 10.1016/j.srs.2024.100172.

[36] A. Clarke *et al.*, "Integrating Climate and Satellite Data for Multi-Temporal Pre-Harvest Prediction of Head Rice Yield in Australia," pp. 1–20, 2024.

Deep Neural Network Architectures for Real-Time Drone Detection in Aerial Surveillance Systems: A Comparative Evaluation of YOLOv7, YOLOv8, and YOLOv11

Nazerke Abylay[1], Elvira Kadylbekkyzy[2], Bakhytzhan Kulambayev[3],
Bayan Moldakalykova[4], Zharkyn Bimoldina[5], Saken Mambetov[6]*, Kuanysh Dossanbek[7]
Department of Telecommunication Engineering, Energo University, Almaty, Kazakhstan[1, 2]
Higher School of Telecommunications, Turan University, Almaty, Kazakhstan[1, 3]
Higher School of Information Technologies, Turan University, Almaty, Kazakhstan[4, 5, 6]
Department of Computer Engineering, International IT University, Almaty, Kazakhstan[7]

***Abstract*—The proliferation of consumer drones has raised security concerns for critical infrastructure, airports, and urban surveillance, creating a need for reliable real-time detection. This study presents a controlled comparative evaluation of selected YOLO generations — YOLOv7x, YOLOv8x, and YOLOv11x — for single-class drone detection under matched high-level settings: 640 by 640 input, 50 epochs, the AdamW optimizer at an initial learning rate of 0.001, and an effective optimizer batch size of 64. Experiments used the YOLO Drone Detection Dataset (1,012 training and 347 validation images, 369 validation instances). All models were scored with one evaluation implementation applied to raw predictions, removing differences between framework-specific validators. Across five independent training runs, YOLOv8x achieved the highest mean average precision at an IoU threshold of 0.5, 0.912 with a standard deviation of 0.007, ahead of YOLOv11x at 0.892 (0.004) and YOLOv7x at 0.708 (0.008); a Welch t-test gives a p-value of 0.0012 for YOLOv8x against YOLOv11x. A provenance analysis found that 32 of the 347 validation images are near-duplicates of training images; re-scoring on the 315 remaining images lowers every result but preserves the ordering. Zero-shot evaluation on the independently collected DUT Anti-UAV test set, whose targets are predominantly small, reproduces the same ordering while absolute accuracy falls to 0.346 for the best model. Detection accuracy therefore does not increase monotonically with architectural generation, and accuracy measured on this dataset does not transfer to imagery dominated by small targets. YOLOv11x was the smallest model (56.8 million parameters) with the lowest peak inference memory, indicating a trade-off between accuracy and deployment cost.**

Keywords*—UAV detection; drone detection; YOLO; deep learning; real-time object detection; comparative evaluation

I. Introduction

The rapid increase in the number of unmanned aerial vehicles (UAVs) used in civil, commercial, and security applications has raised growing concerns regarding safety, privacy, and airspace management [1]. Drones can be weaponized, used for unauthorized surveillance, or inadvertently interfere with manned aviation, making real-time autonomous detection a critical research and operational requirement [2]. The global drone market is projected to exceed USD 55 billion by 2028, intensifying the need for robust counter-UAV systems capable of operating under strict computational and latency constraints [3].

Traditional UAV detection approaches based on radar, acoustic sensors, and RF signal analysis face inherent limitations, including small target radar cross-sections, high false-positive rates from non-drone aerial objects, and vulnerability to complex electromagnetic environments [4]. Time-frequency CNN-based RF detection has demonstrated promising performance, although its effectiveness decreases at longer operating ranges [5]. Combining surveillance radar with YOLO-based classification can improve detection accuracy, but at the expense of increased system complexity associated with multi-sensor architectures [6]. Recent advances in deep learning, particularly convolutional neural networks (CNNs), have facilitated a shift toward visual and multimodal detection approaches with substantially improved discriminative performance [7]. Machine learning approaches have also been investigated for distinguishing UAVs from visually similar aerial objects such as birds, highlighting the importance of reducing false alarms in automated airspace surveillance systems [8].

This study contributes a controlled benchmarking protocol for comparing generations of YOLO detectors under matched high-level settings; a comparison covering detection accuracy, operating-point behaviour, robustness under image degradation, latency and throughput, with all models scored by one evaluation implementation; an analysis of training behaviour and computational efficiency under a fixed epoch budget; and practical recommendations for detector selection in real-time drone surveillance.

The evolution of object detection architectures from two-stage detectors such as R-CNN [9] and Faster R-CNN [10] to single-stage frameworks including SSD [11] and the YOLO family reflects the field's pursuit of higher accuracy at lower computational cost. For UAV detection, where detection latency

*Corresponding author

directly impacts response effectiveness and onboard power is constrained, this efficiency imperative is particularly acute [12].

Comprehensive reviews of deep learning methods for UAV detection indicate that YOLO-based architectures provide an effective balance between detection accuracy and computational efficiency [13]. The evolution of YOLO architectures from YOLOv1 to YOLOv8 demonstrates a continuous progression in feature extraction, optimization strategies, and detection performance [14]. A review of YOLO-based UAV detection technologies identified multi-scale feature fusion and attention mechanisms as the most influential architectural enhancements for improving detection accuracy [15]. A lightweight UAV detection framework based on GhostConv modules achieved a 40–60% reduction in FLOPs while maintaining competitive detection performance [16]. An efficient lightweight detection network demonstrated a processing speed of 682 FPS while reducing model parameters by 88.5% for UAV imagery analysis [17].

A lightweight object detection network optimized for UAV vision platforms was developed to improve deployment efficiency on resource-constrained devices [18]. CNN-based target detection has also been investigated for UAV reconnaissance imagery, where Faster R-CNN combined with optimized region proposals and image enhancement was used to improve detection under complex backgrounds and varying illumination [19]. An edge-oriented UAV detection framework achieved a favorable balance between accuracy and computational efficiency through an optimized backbone architecture [20]. A lightweight detection network achieved state-of-the-art efficiency for UAV object detection while maintaining competitive accuracy [21].

An enhanced YOLOv8-based architecture reduced the number of parameters by approximately 50% while maintaining an mAP@0.5 value of 98.4% [22]. A lightweight DRBD-YOLOv8 framework achieved higher detection accuracy with approximately half the parameters of YOLOv8n [23]. SEB-YOLOv8s enhanced unauthorized UAV detection through the incorporation of spatial attention mechanisms [24]. Small target-YOLOv5 improved drone aerial imagery analysis using specialized feature enhancement modules [25]. An optimized YOLOv7 framework for small-target detection in UAV aerial imagery combined anchor-box clustering, enhanced multi-scale feature fusion, and a refined detection head to improve detection efficiency on the VisDrone2019 dataset [26]. An improved YOLOv8n-based framework demonstrated competitive performance for lightweight tiny-object detection on the VisDrone2019 dataset [27]. A dual-branch YOLOv8 architecture was further proposed for small-object detection in UAV imagery, integrating enhanced feature extraction and multi-scale feature fusion to improve detection performance on the VisDrone benchmark [28].

Attention mechanisms have emerged as an effective strategy for improving UAV detection performance. EDGS-YOLOv8 combined EMA attention with Ghost convolutions and achieved 97.1% AP with a model size of only 4.23 MB [29]. A modified YOLOv8 architecture demonstrated effective multi-class detection of uncooperative aerial vehicles through attention-enhanced feature extraction [30]. Light-YOLOv5 integrated transformer components and PEAM attention to improve performance on resource-constrained camera platforms [31]. YOLO-LE introduced adaptive feature fusion modules to enhance multi-scale feature sensitivity [32].

Research on hardware-oriented deployment has also progressed considerably. FPGA-based implementations of YOLOv3 and YOLOv8 enabled low-power onboard drone classification [33]. Dedicated FPGA accelerators have been developed to support real-time UAV object detection [34]. Lightweight small-target detection algorithms have been optimized for UAV platforms operating under strict power constraints [35]. LW-YOLOv8 combined CSP-CNN-Transformer fusion with SIoU loss to improve edge-device deployment efficiency [36].

Recent developments have increasingly focused on transformer-hybrid architectures. A Vision Transformer framework incorporating hierarchical attention and dynamic token pruning achieved linear-complexity attention for UAV edge computing applications [37]. LMWP-YOLO improved long-range small-drone detection through multidimensional collaborative attention and network pruning [38]. YOLO-DD employed boundary-aware feature fusion based on YOLOv11 to improve UAV detection in complex environments [39]. FFG-YOLO achieved lightweight UAV detection with only 1.77 million parameters and 5.7 GFLOPs [40]. EDNet introduced scalable model variants and reported a 5.6% improvement in mAP on mobile platforms [41]. A lightweight anti-drone YOLOv8 framework reduced the parameter count by 9.6 million while maintaining competitive detection performance [42]. Evaluation of YOLOv8 on distorted surveillance videos reported mAP values of 92.3% for visible-spectrum imagery and 97.9% for infrared imagery, demonstrating the effectiveness of deep learning-based detectors in multimodal surveillance environments [43]. Recent advances in UAV-based systems increasingly combine edge intelligence, distributed sensing, and machine learning to support autonomous monitoring and real-time data classification [44]. Beyond object detection, recent research has investigated multi-UAV surveillance systems that integrate aerial image stitching to enhance large-scale monitoring and situational awareness capabilities [45].

Despite significant progress in UAV detection research, most studies evaluate a single architecture or use different datasets and experimental configurations. As a result, it remains difficult to compare the performance of different detector generations objectively. A controlled cross-generational comparison under matched training settings and a single evaluation implementation is still lacking in the literature.

Single-stage detectors have been applied to drone detection across successive YOLO generations: Singha and Aydin demonstrated automated drone detection with YOLOv4 [46], Aydin and Singha extended this to YOLOv5 [47], and Hakani and Rawat reported real-time YOLOv9 detection deployed on an NVIDIA Jetson Nano edge device [48], underscoring the practical importance of on-device throughput. The three variants evaluated in the present study span the architectural transition from anchor-based detection (YOLOv7x) to anchor-free decoupled-head detection (YOLOv8x and YOLOv11x) and correspond to three stable, officially maintained extra-large

releases; such stable releases are a prerequisite for applying one training specification across all three. YOLOv9 and YOLOv10 are recent and promising architectures; their evaluation under the same conditions is a direct extension of this benchmark.

To the best of our knowledge, no prior study evaluates YOLOv7x, YOLOv8x and YOLOv11x under one set of matched high-level settings — identical dataset partition, input resolution, optimizer, schedule and hardware — with all three scored by a single evaluation implementation, while jointly reporting detection accuracy, operating-point behaviour, robustness under image degradation and computational cost. The present study is therefore positioned as a controlled cross-generational benchmark rather than a new detector. Its contribution is a like-for-like comparison that holds the training and data factors fixed as far as separately maintained implementations allow, and reports what remains unmatched (Section II-F), yielding a reproducible basis for model selection in real-time UAV surveillance.

II. Materials and Methods

A. Dataset

All experiments used the YOLO Drone Detection Dataset [49], a public benchmark from Kaggle, with its predefined partition, which we did not re-partition: 1,012 training images containing 1,117 annotated drone instances and 347 validation images containing 369 instances. Annotations follow the standard normalized YOLO format for a single class ("drone"). Image resolution varies widely: 150–5,760 px wide by 160–3,840 px high in the training split and 200–6,000 by 159–4,000 px in the validation split. Instance size, computed from each annotation's absolute box area (the normalized label multiplied by that image's own pixel dimensions) and binned by the COCO convention, is strongly skewed towards large objects: in the validation split 13 instances (3.5 %) are small ($< 32^2$ px), 30 (8.1 %) medium (32^2–96^2 px) and 326 (88.3 %) large ($> 96^2$ px); the training split contains 1 small, 48 medium and 1,068 large instances. These figures are computed by the released prepare_data.py script rather than estimated. Images capture diverse UAV types under varying conditions, including daylight, overcast, different altitudes, and urban and rural backgrounds.

The dataset is publicly available on Kaggle [49]. At the time of access, no explicit license statement was provided on the dataset page. Therefore, the authors do not redistribute the dataset or reproduce any original dataset images in this article.

B. Model Architectures

Three extra-large (x) variants from consecutive YOLO generations were evaluated. Table I summarizes their architectural characteristics.

The three variants were selected as widely deployed milestone releases that span the transition from anchor-based to anchor-free detection, each having a stable, officially maintained extra-large variant at the time the experiments were run. The study is therefore a comparison of selected YOLO generations rather than an exhaustive one: YOLOv9 and YOLOv10 were not evaluated. A further consequence of using released checkpoints is that each architecture is initialized from weights produced by its own upstream training pipeline, so initialization is confounded with architecture; this is discussed in the Limitations.

TABLE I. Architectural Characteristics of Evaluated Models

Model	Backbone	Neck	Detection Head	Params (M)	GFLOPs
YOLOv7x	E-ELAN	PAFPN	Anchor-based (IDetect)	70.8	188.0
YOLOv8x	CSPDarknet53 +C2f	PAN-FPN	Anchor-free (Decoupled)	68.1	257.4
YOLOv11x	C3k2	SPPF+PAN	Anchor-free (Decoupled)	56.8	194.8

C. Training Configuration

Table II details the training configuration, which is identical across all three models at the level of these settings. Section II-F documents the implementation-level differences that this level of matching cannot remove. Where an effective optimizer batch size of 64 is reported, it refers to the batch seen by the optimizer update only: gradients are accumulated from physical batches of 8, so batch-normalization statistics are computed over 8 samples and augmentation is sampled per physical batch. The two are not equivalent, and no claim of equivalence is made.

TABLE II. Training Hyperparameters (Identical Across All Models)

Parameter	Value
Input resolution	640 × 640 (letterbox)
Batch size	8 per step, accumulated to a nominal batch of 64
Epochs	50
Optimizer	AdamW ($lr_0 = 0.001$)
Augmentation	Mosaic, HSV jitter, scale ±0.5, flip 0.5
Pre-trained weights	COCO (respective variant)
Confidence threshold	0.25
IoU threshold (NMS)	0.5

D. Experimental Setup

All experiments were conducted in a single standardized hardware and software environment so that performance differences are attributable to the models rather than to the experimental conditions. The computational platform was an NVIDIA GeForce RTX 5070 Ti Laptop GPU with 12 GB of VRAM, running Windows 10, Python 3.14.2, PyTorch 2.11.0, and CUDA 12.8. All models were trained and evaluated with the same data split, optimization settings, learning-rate schedule, and stopping criterion. Because the extra-large variants exceed 12 GB of VRAM at a physical batch of 16, training used a physical batch of 8 with gradient accumulation to an effective optimizer batch size of 64. A sensitivity study over accumulation factors was not performed, since a physical batch of 64 does not fit on this device at this resolution; this is recorded in the Limitations.

E. Reproducibility

YOLOv8x and YOLOv11x were trained with the Ultralytics framework (v8.4.117); YOLOv7x used the official implementation (github.com/WongKinYiu/yolov7, commit a207844b1ce82d204ab36d87d496728d3d2348e7). The software environment was Python 3.14.2, PyTorch 2.11.0+cu128, CUDA 12.8, NumPy 2.4.4, and OpenCV 5.0.0.93; every dependency is pinned in requirements.txt. Training started from the official COCO-pretrained weights: yolov7x.pt, yolov8x.pt and yolo11x.pt. The optimizer was AdamW with an initial learning rate of 1e-3, linear warm-up over the first three epochs, and cosine decay to 1e-5 over 50 epochs. The warm-up bias learning rate is set to 0.0: Ultralytics zeroes this parameter only when the optimizer is selected automatically, and leaving it at its default of 0.1 drives the validation loss to NaN within the first epoch under AdamW, so the protocol is not runnable without specifying it. The reported weights are the best validation checkpoint of each run, not the final epoch. The primary run of Table III uses seed 0, which fixes Python, NumPy, and PyTorch; the stability analysis of Table V uses five independent retrainings per model at seeds 0, 1, 2, 3, and 4, and the individual results of every run are reported there. The exact training and validation commands, configuration files, environment specification, per-run result files, and trained weights are available at https://github.com/sakenski-afk/drone-detection-benchmark and archived at Zenodo (https://doi.org/10.5281/zenodo.21955344), with the trained weights provided as release assets in the repository and SHA-256 checksums linking each checkpoint to the results computed from it.

The 50-epoch schedule is a fixed training budget applied identically to all three models rather than a demonstrated convergence point. As Fig. 1 shows, validation mAP@0.5 was still rising at the end of the schedule for every model, by +0.028 (YOLOv7x), +0.023 (YOLOv8x) and +0.021 (YOLOv11x) over the final ten epochs, with best checkpoints at epochs 35, 44 and 39 respectively. The comparison is therefore valid as a like-for-like comparison at a fixed budget, while absolute values would likely improve under a longer schedule.

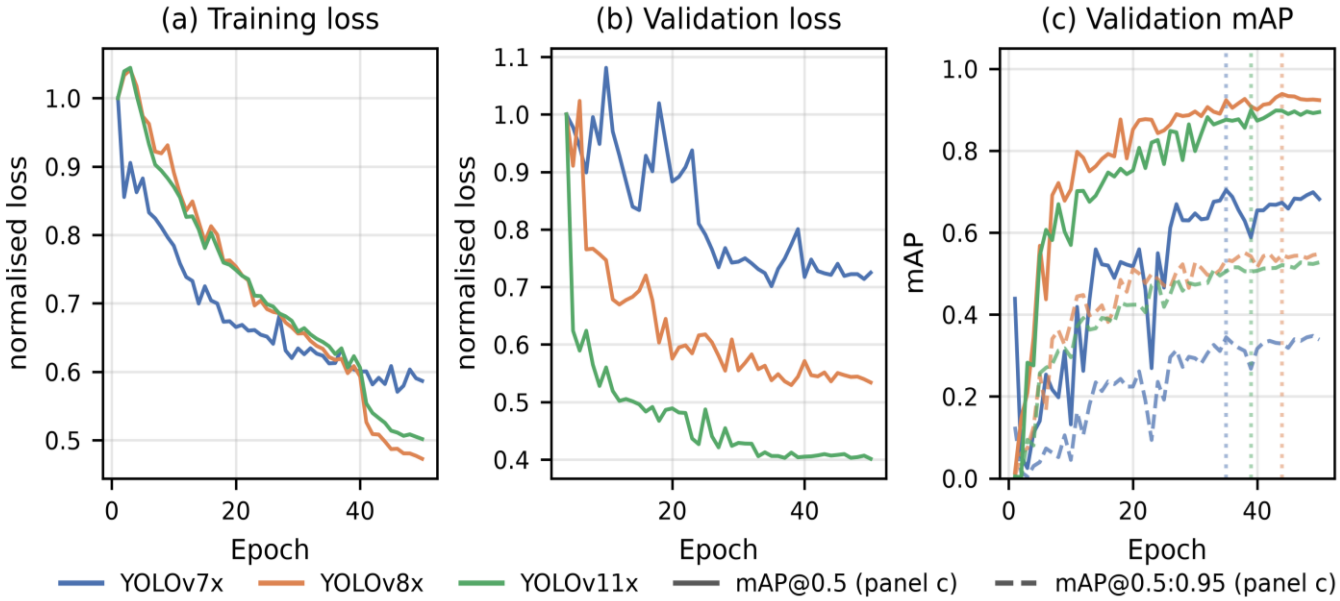


Fig. 1. Training loss, validation loss and validation mAP per epoch for the three architectures (seed 0). The two frameworks report different loss terms, so losses are normalized and only their shapes are comparable: training loss to its epoch-1 value, validation loss to its epoch-4 value, the first three epochs being warm-up. Dotted vertical lines in panel (c) mark the best validation checkpoint of each model. Validation mAP@0.5 is still rising at epoch 50 for all three architectures.

F. Implementation-Level Differences Between the Frameworks

Matching a protocol across separately maintained implementations is possible at the level of settings but not at the level of pipelines, and the residual differences are not hypothetical. The YOLOv7 repository provides no AdamW option: its --adam flag constructs torch. optim.Adam, so obtaining the optimizer of Table II required patching that call site, and the patch is recorded in the released repository. The loss functions differ in kind: YOLOv7x uses OTA assignment with objectness, box, and classification terms, while YOLOv8x and YOLOv11x use a decoupled anchor-free head with distribution focal loss, so loss values are not comparable across frameworks. Augmentation, exponential moving average, and pre- and post-processing defaults also differ where the nominal hyperparameter names coincide. The one axis on which the comparison is genuinely identical is evaluation: all three models are scored by a single implementation applied to raw predictions, verified against the Ultralytics average-precision computation in the released test suite. We therefore describe this study as a comparison under matched high-level settings rather than under an identical training pipeline.

G. Evaluation Metrics

Performance was evaluated using precision, recall, F1-score and mean average precision at IoU = 0.5 (mAP@0.5) and IoU = 0.5:0.95 (mAP@0.5:0.95). All metrics for all three models are computed by a single implementation applied to raw predictions, so that the comparison does not inherit differences between the

frameworks' own validators; the average-precision computation is verified against the Ultralytics implementation to floating-point equality in the released test suite. Predictions are matched to ground truth greedily in descending confidence order, a match requiring IoU ≥ 0.5. Precision, recall, F1, and the confusion matrices are all reported at the operating confidence threshold of 0.25: true positives are matched predictions, false negatives are ground-truth instances left unmatched, so that TP + FN equals the 369 annotated validation instances exactly, and false positives are predictions matched to no instance. The background–background cell of the confusion matrix is undefined and is reported as such: an image contains an unbounded number of candidate windows, so object detection provides no finite set of negatives. For the same reason, ROC-AUC is not reported, since a false-positive rate would require a denominator the data does not determine; free-response ROC (FROC) is reported instead, plotting sensitivity against false positives per image, a rate whose denominator is the number of evaluation images. Robustness was assessed over five independent applications of each degradation to the trained model: additive Gaussian noise with zero mean and $\sigma = 0.05$ in normalized intensity, linear motion blur with a 15 px kernel at a random orientation, and low-light simulation scaling pixel intensities by 0.4 with $\gamma = 1.5$. This resamples noise realizations only and involves no retraining; it is distinct from the retraining variability reported as the standard deviation over five seeds.

III. Results

A. Overall Detection Performance

Table III presents the detection performance of the evaluated models on the validation partition. YOLOv8x achieved the highest detection performance, with a precision of 0.895, recall of 0.902, F1-score of 0.899, mAP@0.5 of 0.921, and mAP@0.5:0.95 of 0.540, ahead of YOLOv11x (0.874, 0.846, 0.860, 0.891, and 0.507). YOLOv7x under the shared protocol reached mAP@0.5 = 0.716; retrained with its own repository's SGD recipe, it reached 0.853, and both are reported because the difference is a property of the matched protocol rather than of the architecture (Section II-F). Precision and recall are taken at the operating threshold of 0.25, consistent with the confusion matrices; Table VII shows that the reported ordering is unchanged when each model is instead scored at its own best-F1 threshold. Because the last-epoch checkpoint involves no selection, the mean of the per-seed differences between it and the reported best-validation checkpoint bounds the optimism introduced by selecting on this partition. Averaged over the five seeds that difference is +0.0056 for YOLOv7x, +0.0058 for YOLOv8x, and −0.0011 for YOLOv11x mAP@0.5; each figure is the mean of the per-seed differences computed before rounding, and the individual runs are reported in the released result files.

TABLE III. Comparative Detection Performance on the Validation Set (347 Images, 369 Instances)

Model	Precision	Recall	F1-score	mAP@0.5	mAP@0.5:0.95	Latency (ms)	FPS
YOLOv7x	0.805	0.672	0.733	0.716	0.357	16.12	62.1
YOLOv8x	0.895	0.902	0.899	0.921	0.540	17.78	56.2
YOLOv11x	0.874	0.846	0.860	0.891	0.507	16.39	61.0
YOLOv7x (native SGD)	0.774	0.799	0.787	0.853	0.450	16.12	62.1

All values are on the validation partition (347 images, 369 instances). Precision, recall, and F1 at the operating confidence threshold of 0.25 with IoU ≥ 0.5 matching. Lat. = inference latency, measured on the GPU of Section II-D and not comparable with other hardware. The final row reports YOLOv7x trained with its own repository's SGD recipe instead of the shared AdamW protocol (Section II-F).

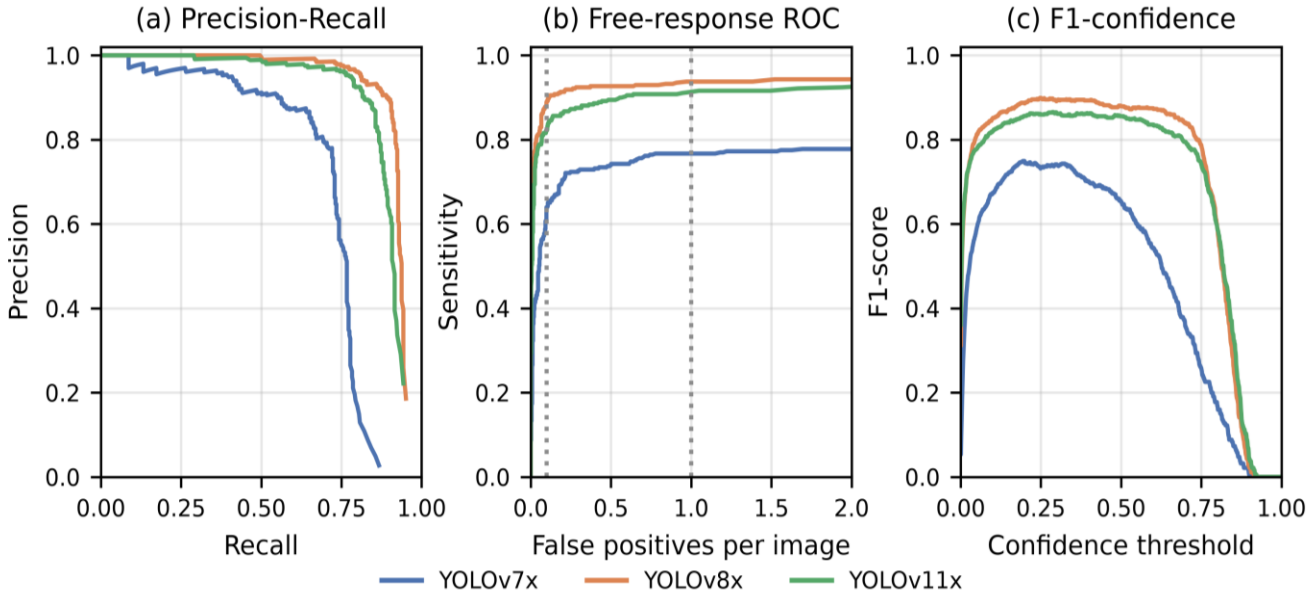


Fig. 2. Precision–recall curves, free-response ROC (FROC) curves, and F1-confidence curves.

B. Precision–Recall, FROC and F1 Analysis

Fig. 2 shows the precision–recall curves, the free-response ROC curves, and the F1-confidence curves for the three architectures. YOLOv8x attains higher precision across most recall levels and the highest sensitivity at every false-positive-per-image operating point examined. At 0.1, 0.25, 0.5, 1.0 and 2.0 false positives per image its sensitivity is 88.9 %, 91.9 %, 92.7 %, 93.8 % and 94.3 %, against 82.1 %, 87.3 %, 89.4 %, 91.3 % and 92.4 % for YOLOv11x and 64.0 %, 72.4 %, 74.0 %, 76.7 % and 77.8 % for YOLOv7x. The F1-confidence curves peak at similar confidence values for the two anchor-free models, indicating comparable confidence calibration.

C. Confusion Matrix Analysis

Fig. 3 presents the confusion matrices at the operating threshold of 0.25. YOLOv8x achieved the highest number of true positives (333 of 369 instances) with 39 false positives, against 312 true positives and 45 false positives for YOLOv11x and 248 true positives and 60 false positives for YOLOv7x. In every matrix, the true positives and false negatives sum to the 369 annotated validation instances; the background–background cell is undefined for object detection and is marked as such rather than filled with a count.

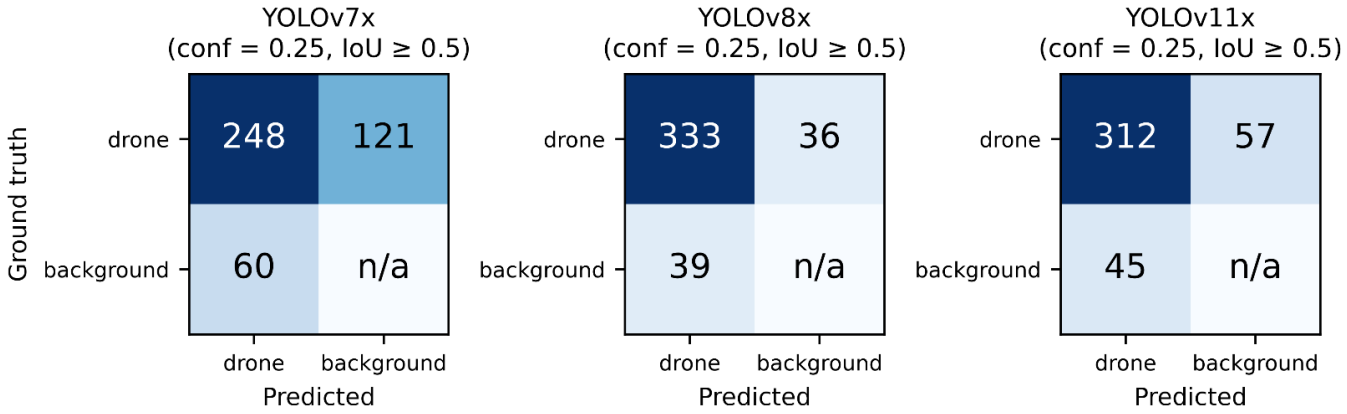


Fig. 3. Confusion matrices for each model on the validation partition, at the operating confidence threshold of 0.25 with IoU ⩾ 0.5 matching. True positives and false negatives sum to the 369 annotated instances in every matrix. The background − background cell is undefined for object detection and is marked as such.

D. Accuracy–Speed Trade-off

The trade-off between detection accuracy and inference speed is illustrated in Fig. 4. YOLOv8x attains the highest mAP@0.5 but the lowest throughput of the three, consistent with its higher computational cost, while YOLOv11x sits between the two on accuracy with the smallest parameter count. No single model dominates the accuracy–efficiency frontier on this dataset, and the ordering by accuracy does not follow the ordering by architectural generation.

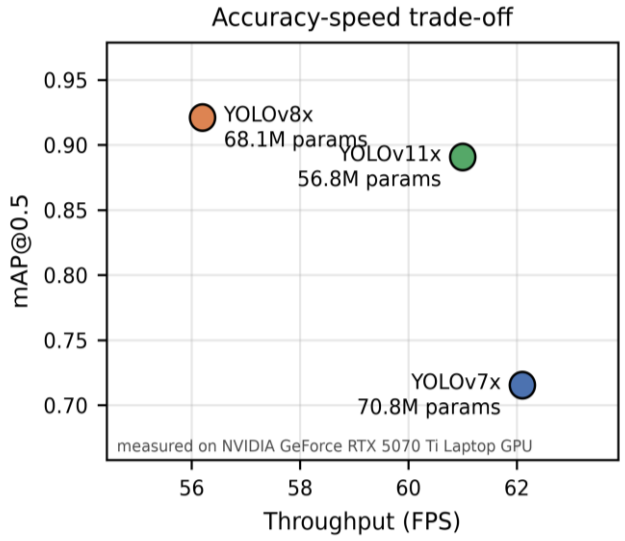


Fig. 4. Accuracy–speed trade-off (mAP@0.5 vs FPS).

E. Computational Efficiency

Table IV summarizes the computational complexity of the evaluated models, measured on the single-class (nc = 1) fused models. YOLOv11x required 56.8 million parameters, a 19.8 % reduction relative to YOLOv7x, and is the smallest of the three, while YOLOv8x carries the highest computational cost at 257.4 GFLOPs. Latency and throughput were measured on the GPU described in Section II-D as the mean of 100 forward passes at batch size 1 after 10 warm-up iterations, and track the GFLOPs ordering. These measurements are hardware-bound and are not comparable with figures obtained on other devices.

TABLE IV. COMPUTATIONAL EFFICIENCY COMPARISON

Model	Params(M)	GFLOPs	Layers	Lat.(ms)	FPS
YOLOv7x	70.8	188.0	362	16.12	62.1
YOLOv8x	68.1	257.4	113	17.78	56.2
YOLOv11x	56.8	194.8	191	16.39	61.0

F. Stability and Robustness Analysis

Reliability was assessed in two ways that do not carry the same evidential weight, and are therefore reported separately. Table V gives the mean and standard deviation of mAP@0.5 over five independent training runs, which differ only in random seed. Table VI gives the mean over five independent applications of each image degradation to the single seed-0 model; that procedure resamples noise realizations and blur orientations without retraining, and so says nothing about training variability. Across the five seeds, the ranges of

YOLOv8x (0.9035–0.9210) and YOLOv11x (0.8861–0.8968) do not overlap, so the ordering between them is not an artefact of a single seed.

TABLE V. STABILITY ACROSS FIVE INDEPENDENT TRAINING RUNS

Model	Baseline mAP@0.5 (seed 0)	Mean ± SD over 5 independent training runs
YOLOv7x	0.716	0.708 ± 0.008
YOLOv8x	0.921	0.912 ± 0.007
YOLOv11x	0.891	0.892 ± 0.004
Stability of the evaluated models. Each value is mAP@0.5 on the validation partition from an independent training run; the five runs differ only in random seed.		

TABLE VI. ROBUSTNESS UNDER IMAGE DEGRADATION (SINGLE MODEL, FIVE DEGRADATION REALIZATIONS)

Model	Baseline mAP@0.5	Gaussian noise	Motion blur	Low light
YOLOv7x	0.716	0.598	0.645	0.604
YOLOv8x	0.921	0.906	0.769	0.853
YOLOv11x	0.891	0.828	0.783	0.858
Each value is the mean over five applications of the degradation to the seed-0 model. No retraining is involved, so these columns carry no information about training variability.				

The robustness results are consistent with the detection performance reported above: the anchor-free models retain a higher mAP@0.5 than YOLOv7x under all three degradations, and at the operating threshold they produce fewer false positives (39 for YOLOv8x and 45 for YOLOv11x against 60 for YOLOv7x).

G. Sensitivity to the Operating Threshold

Precision, recall, and F1 in Table III are reported at a single confidence threshold of 0.25 shared by all models, which need not coincide with any one model's best operating point. Table VII therefore scores each model both at the shared threshold and at its own best-F1 threshold. The shared threshold happens to sit almost exactly at YOLOv8x's optimum and costs YOLOv7x 0.017 and YOLOv11x 0.006 F1, but the ordering is identical under both, and under mAP@0.5 and mAP@0.5:0.95, which integrate over the confidence sweep and cannot depend on the choice of operating point at all.

TABLE VII. DETECTION PERFORMANCE AT THE SHARED AND AT EACH MODEL'S OWN BEST-F1 THRESHOLD

Model	F1 at 0.25	Best F1	At threshold	Cost of sharing	mAP@0.5
YOLOv7x	0.733	0.749	0.199	-0.017	0.716
YOLOv8x	0.899	0.899	0.251	-0.000	0.921
YOLOv11x	0.860	0.865	0.289	-0.006	0.891
The ordering YOLOv8x > YOLOv11x > YOLOv7x holds under all four criteria.					

H. Independence of the Validation Partition

The train and validation partitions are those distributed with the dataset and were not created by us, so their independence cannot be assumed. Three signals were used to test it: an exact hash of the file bytes, a 64-bit perceptual hash, and cosine similarity between ImageNet-pretrained ResNet-50 features. An ImageNet model is used deliberately, since features from a model trained on this dataset would beg the question. Table VIII reports the result.

The partitions are not independent. 15 validation images are byte-identical to training images, and several further pairs have a perceptual-hash distance of zero with a cosine similarity of 1.000. Taking the union of the three signals, 32 of 347 validation images (9.2 %) cannot be treated as independent evidence.

Every model was therefore re-scored on the 315 remaining images (Table IX). All results fall by 0.011 to 0.027 mAP@0.5, so the figures obtained on the full partition are optimistic. The ordering is unaffected: the per-seed ranges remain disjoint, YOLOv8x spanning 0.8923–0.9105 against 0.8729–0.8857 for YOLOv11x, and the difference remains significant (Welch t-test, $p = 0.00117$). The contamination inflated the measurements; it did not produce the result. The list of affected images is released with the code.

TABLE VIII. SIGNALS OF CONTAMINATION BETWEEN THE TRAINING AND VALIDATION PARTITIONS

Signal	Criterion	Validation images
Exact duplicate	identical SHA-256 of the file bytes	15
Perceptual hash	64-bit DCT hash within 5 bits	32
Embedding similarity	ResNet-50 cosine at least 0.95	20
Union of all three	—	32 of 347
Each row counts validation images matching at least one training image under that criterion.		

TABLE IX. DETECTION PERFORMANCE RE-SCORED ON THE LEAKAGE-FREE SUBSET

Model	Full partition (347 images)	Leakage-free (315 images)	Difference
YOLOv7x	0.7079 ± 0.0082	0.6813 ± 0.0089	+0.0266
YOLOv8x	0.9115 ± 0.0067	0.9002 ± 0.0075	+0.0113
YOLOv11x	0.8924 ± 0.0042	0.8790 ± 0.0052	+0.0134
Mean ± standard deviation of mAP@0.5 over five independent training runs.			

I. Cross-Dataset Generalization

Whether the ordering established above is a property of the architectures or of this dataset can only be decided on data the models have not seen. Every trained checkpoint — three architectures, five seeds each — was therefore evaluated, without retraining or fine-tuning, on the detection test split of the DUT Anti-UAV dataset [50]. That set was chosen because its target-scale distribution is close to the inverse of ours (Table X), which makes it a test of exactly the regime this dataset cannot represent. A perceptual-hash check confirms it shares no near-duplicate imagery with our training partition.

The ordering transfers unchanged (Table XI): YOLOv8x leads YOLOv11x, and YOLOv7x remains far behind, with the difference between the two leading models significant over the

five runs (Welch t-test, p = 0.00206, Cohen's d = 2.83). Unlike in our own data, the per-seed ranges overlap slightly here, so the lead is significant on means rather than disjoint run by run.

TABLE X. INSTANCE-SIZE DISTRIBUTION OF THE EVALUATION SETS

Instance size (COCO area convention)	This study (validation)	DUT Anti-UAV (test)
Small, below 32^2 px	13 (3.5 %)	848 (37.8 %)
Medium, 32^2–96^2 px	30 (8.1 %)	836 (37.2 %)
Large, above 96^2 px	326 (88.3 %)	561 (25.0 %)
Images/instances	347 / 369	2200 / 2245
Sizes follow the COCO area convention, computed from each annotation against its own image dimensions.		

Absolute accuracy does not transfer. The best model retains 38 % of its in-dataset mAP@0.5 and YOLOv7x only 19 %, on a set where 37.8 % of instances are small against 3.5 % here. Fig. 5 places the three evaluation sets side by side: removing the contaminated images lowers every model slightly, while the change of domain lowers them by more than half, and the ordering survives both. Accuracy measured on this dataset therefore predicts the ranking of these architectures on other data, but not their accuracy on it, and in particular says little about performance on small and distant drones.

TABLE XI. ZERO-SHOT CROSS-DATASET PERFORMANCE, MAP@0.5

Model	This dataset	DUT Anti-UAV (zero-shot)	Retained
YOLOv7x	0.708 ± 0.008	0.132 ± 0.020	19 %
YOLOv8x	0.912 ± 0.007	0.346 ± 0.007	38 %
YOLOv11x	0.892 ± 0.004	0.327 ± 0.007	37 %
Five independent training runs per model in both columns. Zero-shot means no retraining or fine-tuning on the external set.			

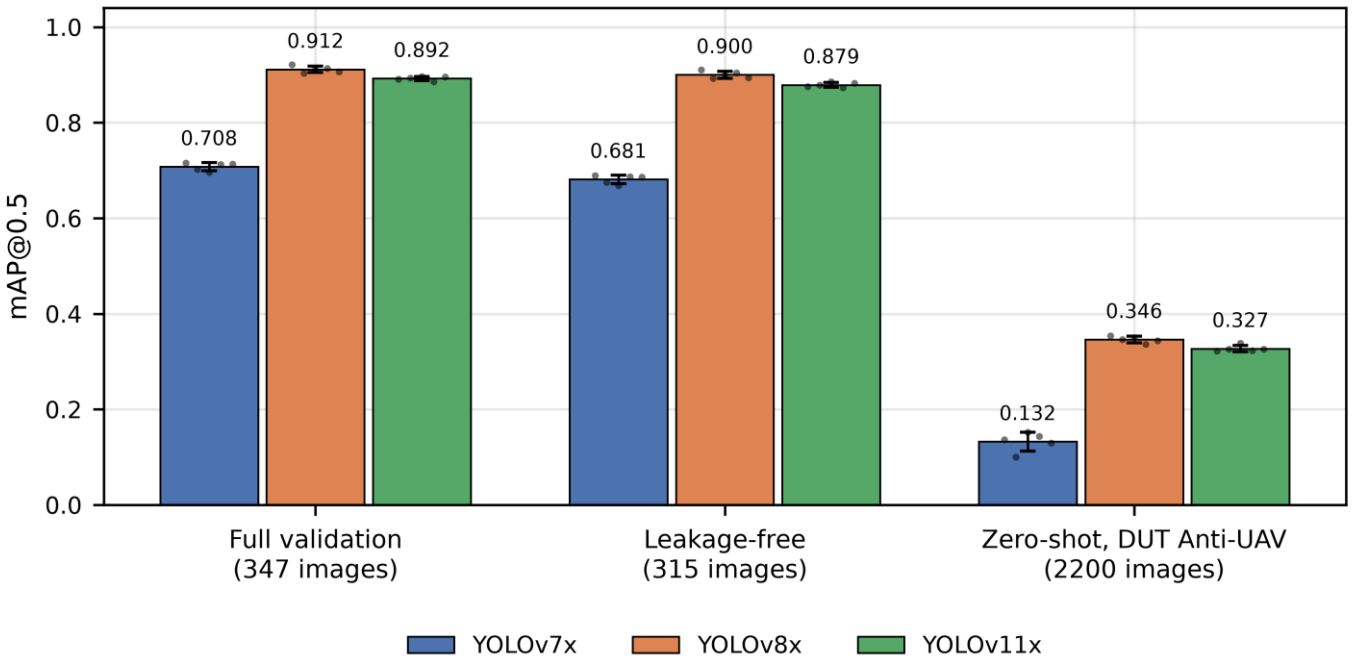


Fig. 5. Detection accuracy across the three evaluation sets. Bars are the mean over five independent training runs, error bars are one standard deviation, and points are the individual runs. Removing the contaminated validation images lowers every model; evaluating zero-shot on an independently collected dataset lowers them much further while preserving the ordering.

IV. DISCUSSION

A. Generational Improvements

Detection accuracy does not increase monotonically with architectural generation on this dataset. YOLOv8x exceeds YOLOv11x by 1.9 pp mAP@0.5 as a mean over five independent training runs (0.912 ± 0.007 against 0.892 ± 0.004), the per-seed ranges do not overlap, and a Welch unequal-variance t-test over the five runs per model gives t = 5.351, p = 0.0012 with Cohen's d = 3.38. Taking per-image AP@0.5 as the experimental unit instead, paired across the 347 validation images, gives a mean difference of −0.0349 in favour of YOLOv8x (paired t-test p = 0.00065; Wilcoxon signed-rank p = 0.00064; Cohen's d_z = −0.185; percentile bootstrap 95 % confidence interval [−0.0553, −0.0154] over 10,000 resamples). Both units of analysis agree in sign and exclude zero. YOLOv11x nevertheless uses 16.6 % fewer parameters than YOLOv8x, so the practical choice between them is a trade-off between accuracy and model size rather than a dominance in either direction. We do not attribute the differences to specific architectural components: isolating such causes would require an ablation study that we did not perform.

B. Operating-Point Behaviour

Because object detection provides no finite set of background samples, we characterize threshold behaviour with free-response ROC rather than ROC-AUC. The FROC curves show that YOLOv8x sustains the highest sensitivity at low false-

positive-per-image rates, which is the regime that matters for surveillance deployment, where each false alarm carries an operational cost. Reducing false positives at a fixed sensitivity remains the main practical difference between the models; identifying which architectural changes produce it is beyond what this benchmark can establish.

C. Edge Deployment Implications

Parameter count alone predicts neither the memory a model needs at inference nor the energy it costs per frame, so both were measured directly (Table XII). Peak inference memory does follow parameter count here: YOLOv11x has the smallest footprint at 369 MiB against 535 MiB for YOLOv7x. Energy per frame does not. Board power differs by under 2 W across the three models, so energy per frame is set by throughput, and YOLOv7x — the largest model — is the cheapest at 1.53 J while YOLOv8x is the most expensive at 1.74 J. The claim that YOLOv11x is the most plausible edge candidate among those evaluated is therefore supported on memory and contradicted on energy. These remain desktop-GPU measurements: they do not establish on-device behaviour on an embedded accelerator, where memory bandwidth and kernel support differ, and on-device deployment is left to future work.

TABLE XII. MEASURED DEPLOYMENT COST AT BATCH SIZE ONE

Model	Parameters (M)	Weights	Activations	Peak memory	Sustained FPS	Energy per frame
YOLOv7x	70.8	288 MiB	247 MiB	535 MiB	53.5	1.532 J
YOLOv8x	68.1	276 MiB	110 MiB	386 MiB	46.3	1.738 J
YOLOv11x	56.8	237 MiB	132 MiB	369 MiB	51.0	1.587 J

Peak memory is measured from a single synchronized forward pass at 640×640. Energy is integrated from board power sampled during sustained inference, net of a 18.1 W idle draw, on the GPU of Section II-D. The throughput column is the sustained rate observed during that energy measurement, with power sampling running concurrently; it is therefore lower than the peak throughput of Table IV, which is the best of repeated 100-pass timings with nothing else running. Energy per frame is computed from the sustained rate, since that is the rate at which the measured power was drawn.

D. Limitations

Several limitations bound these findings. The dataset contains 1,012 training and 347 validation images, and its instances are strongly skewed towards large objects, 88 % of validation instances exceeding 96^2 px; the cross-dataset evaluation quantifies what this costs, with the best model retaining 38 % of its accuracy on a set dominated by small targets. The predefined partitions are not fully independent: 32 of 347 validation images are near-duplicates of training images, and while the leakage-free re-scoring preserves the ordering, all full-partition figures are optimistic by 0.011 to 0.027 mAP@0.5. The dataset provides no independent test partition, so the same data were used for checkpoint and threshold selection; the resulting optimism was measured against the unselected last-epoch checkpoint and is +0.006 mAP@0.5 or smaller. The 50-epoch budget was not run to convergence. Each architecture is initialized from a COCO-pretrained checkpoint produced by its own upstream pipeline, so initialization is confounded with architecture, and the differences cannot be attributed to architectural evolution alone; controlling this would require training all three from scratch, which a dataset of this size cannot support, or checkpoints pretrained under one common recipe, which do not exist for these architectures. Only selected generations were compared: YOLOv9 and YOLOv10 were not evaluated. Only extra-large variants were compared, and results may not generalize to multi-class detection, to adverse conditions such as night or fog, or to nano variants. Gradient accumulation was used to reach an effective optimizer batch size of 64, which does not reproduce a true batch of 64 for batch-normalization statistics or augmentation sampling, and no sensitivity study over accumulation factors was performed because a physical batch of 64 does not fit on the available device at this resolution. Edge measurements are desktop-GPU measurements, and on-device performance was not measured. Finally, matching a training protocol across separately maintained implementations is achievable only at the level of settings, not of pipelines, as Section II-F documents.

V. CONCLUSION

This study presented a controlled comparative evaluation of three generations of YOLO architectures—YOLOv7x, YOLOv8x and YOLOv11x—for real-time drone detection on a publicly available dataset under matched high-level training settings, with all models scored by a single evaluation implementation. On this dataset, YOLOv8x delivered the best detection performance, reaching mAP@0.5 = 0.912 ± 0.007 over five independent training runs against 0.892 ± 0.004 for YOLOv11x and 0.708 ± 0.008 for YOLOv7x, with the YOLOv8x and YOLOv11x per-seed ranges not overlapping (Welch t-test, p = 0.0012). YOLOv11x is the smallest model among those evaluated, at 56.8 million parameters.

The results do not support a monotone improvement across architectural generations. The transition from anchor-based detection in YOLOv7x to anchor-free detection in YOLOv8x and YOLOv11x is associated with a substantial gain under the shared protocol, but the most recent generation did not outperform its predecessor here. The same ordering was obtained zero-shot on the independently collected DUT Anti-UAV test set, so it is not an artefact of this dataset, while absolute accuracy fell to 0.346 for the best model and does not transfer. We further found that a single training protocol does not treat separately maintained implementations equally: YOLOv7x reaches mAP@0.5 = 0.853 under its own repository's SGD recipe against 0.716 under the shared AdamW protocol, so a benchmark that imposes one optimizer on all architectures systematically understates it. We do not attribute the measured differences to individual architectural components, which would require an ablation study we did not perform, and initialization from architecture-specific pretrained checkpoints remains a confounding factor.

From a practical perspective, YOLOv8x is the most accurate of the evaluated models on this dataset, while YOLOv11x offers the smallest parameter count and is therefore the more plausible candidate where memory is the binding constraint; the choice between them is a trade-off rather than a dominance. YOLOv7x remains a viable option for legacy deployments integrated with the established WongKinYiu repository, provided it is trained with that repository's own recipe. Future work should extend this methodology to lightweight variants suitable for edge deployment, to multi-class aerial object detection, to validation under nighttime, adverse weather, and heavy occlusion, and to multi-modal sensor fusion. All experimental code, trained model weights, per-run results, and dataset configurations are publicly available to facilitate reproducibility.

In practical terms, the principal contribution is a reproducible, like-for-like selection guideline rather than a new detector, together with the finding that generational ordering cannot be assumed. Its conclusions are bounded by a modest single-class dataset skewed towards large instances, by the absence of an independent test partition, by a fixed training budget that was not run to convergence, and by a single-GPU evaluation with no on-device measurement. The main open challenges remain the reliable detection of small and distant drones, robustness under adverse weather and nighttime conditions, and verified real-time operation on embedded hardware.

ACKNOWLEDGMENT

This research received no grant from any funding agency.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could influence the results of the work presented in this study

DECLARATION ON GENERATIVE AI

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to assist with language refinement, text editing, grammar improvement, and manuscript organization. The AI tool was not used for data collection, data analysis, model training, experimental evaluation, generation of results, or interpretation of findings. All scientific content, experimental design, analyses, figures, tables, conclusions, and final decisions were developed and verified by the authors. After using the AI tool, the authors carefully reviewed, edited, and validated all generated content and take full responsibility for the content of this publication.

REFERENCES

[1] Yasmine, G.; Ahmed, E.; Medromi, H. Real-time lightweight drone detection model: Fine-grained Identification of four types of drones based on an improved Yolov7 model. Neurocomputing 2024, 596, 127941. https://doi.org/10.1016/j.neucom.2024.127941

[2] Mun, J.; Sohn, C. Technology trends in deep learning for real-time object detection in drone environment. J. Adv. Mil. Stud. 2023, 6, 181–196. https://doi.org/10.37944/jams.v6i2.220

[3] Lykou, G.; Moustakas, D.; Gritzalis, D. Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies. Sensors 2020, 20, 3537. https://doi.org/10.3390/s20123537

[4] H. Haifawi, F. Fioranelli, A. Yarovoy and R. van der Meer, "Drone Detection & Classification with Surveillance 'Radar On-The-Move' and YOLO," 2023 IEEE Radar Conference (RadarConf23), San Antonio, TX, USA, 2023, pp. 1-6, https://doi.org/10.1109/RadarConf2351548.2023.10149588

[5] S. Mandal and U. Satija, "Time–Frequency Multiscale Convolutional Neural Network for RF-Based Drone Detection and Identification," in IEEE Sensors Letters, vol. 7, no. 7, pp. 1-4, July 2023, Art no. 7003304, https://doi.org/10.1109/LSENS.2023.3289145

[6] K. P. Solanki, P. Verma and S. Dara, "Evaluating YOLO V8 Models for Efficient Drone Detection in Anti-UAV Systems," 2025 6th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 2025, pp. 1-6, https://doi.org/10.1109/RAIT65068.2025.11089241

[7] Averina, M.D., Levanova, O.A., Grushevskaya, D.V. et al. UAV Detection Using Neural Networks. Aut. Control Comp. Sci. 59, 910–919 (2025). https://doi.org/10.3103/S0146411625700270

[8] Alqaraleh, M., Alzboon, M. S., & Al-Batah, M. S. (2024). Skywatch: Advanced Machine Learning Techniques for Distinguishing UAVs from Birds in Airspace Security. International Journal of Advanced Computer Science and Applications, 15(11). https://doi.org/10.14569/IJACSA.2024.01511104

[9] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587. https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf

[10] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, https://doi.org/10.1109/TPAMI.2016.2577031

[11] Liu, W. et al. (2016). SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2

[12] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao; YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464-7475. https://doi.org/10.48550/arXiv.2207.02696

[13] Al-lQubaydhi, N., Alenezi, A., Alanazi, T., Senyor, A., Alanezi, N., Alotaibi, B., ... & Hariri, S. (2024). Deep learning for unmanned aerial vehicles detection: A review. Computer Science Review, 51, 100614. https://doi.org/10.1016/j.cosrev.2023.100614

[14] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO," in IEEE Access, vol. 12, pp. 42816-42833, 2024, https://doi.org/10.1109/ACCESS.2024.3378568

[15] Chen, C., Zheng, Z., Xu, T., Guo, S., Feng, S., Yao, W., & Lan, Y. (2023). YOLO-Based UAV Technology: A Review of the Research and Its Applications. Drones, 7(3), 190. https://doi.org/10.3390/drones7030190

[16] Cao, J., Bao, W., Shang, H., Yuan, M., & Cheng, Q. (2023). GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. Remote Sensing, 15(20), 4932. https://doi.org/10.3390/rs15204932

[17] Li, H., Ma, J., & Zhang, J. (2025). ELNet: An Efficient and Lightweight Network for Small Object Detection in UAV Imagery. Remote Sensing, 17(12), 2096. https://doi.org/10.3390/rs17122096

[18] Chen, N., Li, Y., Yang, Z. et al. LODNU: lightweight object detection network in UAV vision. J Supercomput 79, 10117–10138 (2023). https://doi.org/10.1007/s11227-023-05065-x

[19] Na, L. (2024). CNN-Based Salient Target Detection Method of UAV Video Reconnaissance Image. International Journal of Advanced Computer Science and Applications, 15(9). https://doi.org/10.14569/IJACSA.2024.0150909

[20] X. Min, W. Zhou, R. Hu, Y. Wu, Y. Pang and J. Yi, "LWUAVDet: A Lightweight UAV Object Detection Network on Edge Devices," in IEEE Internet of Things Journal, vol. 11, no. 13, pp. 24013-24023, 1 July 2024, https://doi.org/10.1109/JIOT.2024.3388045

[21] Hua, W., Chen, Q. & Chen, W. A new lightweight network for efficient UAV object detection. Sci Rep 14, 13288 (2024). https://doi.org/10.1038/s41598-024-64232-z

[22] Zhao, Y., Wang, L., Lei, G., Guo, C., & Ma, Q. (2024). Lightweight UAV Small Target Detection and Perception Based on Improved YOLOv8-E. Drones, 8(11), 681. https://doi.org/10.3390/drones8110681

[23] Jiang, P., Yang, X., Wan, Y., Zeng, T., Nie, M., & Liu, Z. (2024). DRBD-YOLOv8: A Lightweight and Efficient Anti-UAV Detection Model. Sensors, 24(22), 7148. https://doi.org/10.3390/s24227148.

[24] Fang, A., Feng, S., Liang, B., & Jiang, J. (2024). Real-Time Detection of Unauthorized Unmanned Aerial Vehicles Using SEB-YOLOv8s. Sensors, 24(12), 3915. https://doi.org/10.3390/s24123915

[25] Zhou, J., Su, T., Li, K., & Dai, J. (2024). Small Target-YOLOv5: Enhancing the Algorithm for Small Object Detection in Drone Aerial Imagery Based on YOLOv5. Sensors, 24(1), 134. https://doi.org/10.3390/s24010134

[26] Liu, Y., Chen, S., & Luo, L. (2023). Optimized YOLOv7 for Small Target Detection in Aerial Images Captured by Drone. International Journal of Advanced Computer Science and Applications, 14(9). https://doi.org/10.14569/IJACSA.2023.0140909

[27] Yue, M., Zhang, L., Huang, J., & Zhang, H. (2024). Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images. Drones, 8(7), 276. https://doi.org/10.3390/drones8070276

[28] Tan, Y., Xu, B., Sun, J., Xu, C., Pan, W., Dai, S., & Liu, H. (2025). DBYOLOv8: Dual-Branch YOLOv8 Network for Small Object Detection on Drone Image. International Journal of Advanced Computer Science and Applications, 16(1). https://doi.org/10.14569/IJACSA.2025.01601124

[29] Huang, M., Mi, W., & Wang, Y. (2024). EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model. Drones, 8(7), 337. https://doi.org/10.3390/drones8070337

[30] P. Varma, K. Singh, A. Das, D. K. Giri and P. N. Dwivedi, "Modified YOLOv8 for Real-Time Multi-Class Uncooperative Aerial Vehicle Detection and Classification," 2025 IEEE 21st International Conference on Automation Science and Engineering (CASE), Los Angeles, CA, USA, 2025, pp. 3010-3015, https://doi.org/10.1109/CASE58245.2025.11163927

[31] J. Han, R. Cao, A. Brighente and M. Conti, "Light-YOLOv5: A Lightweight Drone Detector for Resource-Constrained Cameras," in IEEE Internet of Things Journal, vol. 11, no. 6, pp. 11046-11057, 15 March15, 2024, https://doi.org/10.1109/JIOT.2023.3329221

[32] Chen, Z., Zhang, Y., & Xing, S. (2025). YOLO-LE: A Lightweight and Efficient UAV Aerial Image Target Detection Model. Computers, Materials & Continua, 84(1). http://dx.doi.org/10.32604/cmc.2025.065238

[33] Di Cecio, G., Manco, A., & Gigante, G. (2024, March). On-board drone classification with Deep Learning and System-on-Chip implementation. In Journal of Physics: Conference Series (Vol. 2716, No. 1, p. 012059). IOP Publishing. https://doi.org/10.1088/1742-6596/2716/1/012059

[34] G. Li, J. Zhang, M. Zhang and H. Corporaal, "An Efficient FPGA Implementation for Real-Time and Low-Power UAV Object Detection," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 1387-1391, https://doi.org/10.1109/ISCAS48785.2022.9937449

[35] Lv, Y., Tian, B., Guo, Q., & Zhang, D. (2025). A Lightweight Small Target Detection Algorithm for UAV Platforms. Applied Sciences, 15(1), 12. https://doi.org/10.3390/app15010012

[36] Ma, F., Zhang, R., Zhu, B. et al. A lightweight UAV target detection algorithm based on improved YOLOv8s model. Sci Rep 15, 15352 (2025). https://doi.org/10.1038/s41598-025-00341-7

[37] Zhu, W., Chen, K. Real-time object detection for unmanned aerial vehicles based on vision transformer and edge computing. Sci Rep 16, 6814 (2026). https://doi.org/10.1038/s41598-026-37938-5

[38] Zhou, S., Yang, L., Liu, H. et al. Improved YOLO for long range detection of small drones. Sci Rep 15, 12280 (2025). https://doi.org/10.1038/s41598-025-95580-z

[39] Zheng, C., Liu, L., Fu, Q. et al. YOLO-DD: a lightweight framework for UAV detection in complex environments via boundary-aware fusion. EURASIP J. Adv. Signal Process. 2025, 44 (2025). https://doi.org/10.1186/s13634-025-01253-4

[40] Wang, T., Yang, S., Wan, M., & Liu, Y. (2025). FFG-YOLO: Improved YOLOv8 for Target Detection of Lightweight Unmanned Aerial Vehicles. Applied System Innovation, 8(4), 109. https://doi.org/10.3390/asi8040109

[41] Z. Song, Y. Zhang and A. Ebayyeh, "EDNet: Edge-Optimized Small Target Detection in UAV Imagery - Faster Context Attention, Better Feature Fusion, and Hardware Acceleration," 2024 IEEE Smart World Congress (SWC), Nadi, Fiji, 2024, pp. 829-838, https://doi.org/10.1109/SWC62898.2024.00141

[42] Y. Benchen and D. Xuan, "Improving the lightweight anti drone target detection algorithm for YOLOv8," 2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Hangzhou, China, 2024, pp. 402-408, https://doi.org/10.1109/ICCASIT62299.2024.10827908

[43] Zainudin, A. Z. B., Abdul Karim, H. B., & Aldahoul, N. (2024). Drone detection and tracking in distorted surveillance video. 8th International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia. IEEE. https://doi.org/10.1109/ICSIPA62061.2024.10686164

[44] Mambetov, S.; Nurpeissova, D.; Taissariyeva, K.; Tleuberdiyeva, G.; Mukanova, Z.; Kulambayev, B.; Moshkalov, A.; Skakova, A. Integrated IoT–UAV Architecture for Three-Dimensional Electromagnetic Radiation Monitoring and Intelligent Source Classification. *Electronics* 2026, 15, 1941. https://doi.org/10.3390/electronics15091941

[45] A. El-alami, Y. Nadir, and K. Mansouri, "An Efficient Geometric Transformation-Based Approach for Multi-UAV Image Stitching", *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 25507–25513, Aug. 2025.

[46] Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. Drones 2021, 5(3), 95. https://doi.org/10.3390/drones5030095

[47] Aydin, B.; Singha, S. Drone Detection Using YOLOv5. Eng 2023, 4(1), 416–433. https://doi.org/10.3390/eng4010025

[48] Hakani, R.; Rawat, A. Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano. Drones 2024, 8(11), 680. https://doi.org/10.3390/drones8110680

[49] YOLO Drone Detection Dataset. Kaggle. https://www.kaggle.com/datasets/muki2003/yolo-drone-detection-dataset (accessed 13 April 2026).

[50] Zhao, J., Zhang, J., Li, D., & Wang, D. (2022). Vision-Based Anti-UAV Detection and Tracking. IEEE Transactions on Intelligent Transportation Systems, 23(12), 25323–25334. https://doi.org/10.1109/TITS.2022.3177627

Next-Gen Agriculture: Enhancing Animal Welfare Through IoT, Edge Devices, and Artificial Intelligence

Benlahsiniya Maroua, Ait Abdelouahid Rachida, Marzak Abdelaziz
Department of Mathematics and Computer Science-Artificial Intelligence and Systems Laboratory-
Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca, Casablanca, Morocco

***Abstract*—Agriculture has always been a field of innovation, evolving alongside technological advancements to boost productivity and address sector-specific challenges. Today, the Internet of Things (IoT), edge computing, and artificial intelligence (AI) are revolutionizing farming practices, particularly in livestock management. These technologies enable real-time monitoring of animal health, facilitate early disease detection, and optimize decision-making. Beyond immediate benefits, they contribute to more sustainable agriculture and improved animal welfare, issues of growing concern to consumers. However, adoption faces several hurdles, notably regarding cost, data management, and connectivity. Solutions are emerging to overcome these challenges, such as advanced technology integration, remote sensing, and improved digital infrastructure. In this article, we propose a hybrid Edge-Fog-Cloud architecture for the intelligent monitoring of animal welfare, based on a simulation using synthetic data (100,000 observations covering 100 simulated animals). In addition to standard classification metrics (accuracy, precision, recall, F1-score), the article presents a comprehensive factorial ablation study to determine the marginal contribution of each method, as well as indicators related to event-level detection (detection delay, alert fragmentation, false alert rate) and model performance in terms of resource usage and latency; the aim is to highlight the advantages of the proposed architecture and justify it with quantified, reproducible evidence, rather than relying solely on aggregate accuracy figures. The results presented here constitute a proof of concept based on simulated data, not a demonstration of agricultural performance under real-world conditions with live animals.**

Keywords*—Smart farming; animal welfare; IoT; edge computing; artificial intelligence; connected livestock farming

I. Introduction

Agriculture has always been at the forefront of technological innovation, adapting to advancements to improve production and address sector challenges. Today, with the advent of the Internet of Things (IoT), edge devices, and artificial intelligence (AI), a new agricultural revolution is underway. These technologies offer unprecedented opportunities to transform livestock farming practices and ensure animal welfare, an aspect often sidelined in the past.

Thanks to IoT and AI, farmers can monitor animal health in real time, detect diseases early, and even anticipate needs and behaviors [2], [3]. Furthermore, edge devices enable on-site data processing, thereby reducing latency and optimizing decision-making [4]. These advancements go beyond improving animal health; they also help increase farm efficiency, ensure more sustainable production, and meet growing consumer expectations regarding animal welfare.

However, these technologies present major challenges (detailed in Section I) that still block their large-scale adoption. Following a state-of-the-art review of key technologies (Sections II through VI), this article proposes a hybrid Edge-Fog-Cloud architecture for monitoring animal and agricultural machinery to continuously collect data on the environment and welfare (Section VII), validates it through simulation including a factorial ablation study, event-level metrics, and a resource/latency characterization (Sections VIII and IX), discusses its limitations (Section X), and outlines prospects for embedded deployment (Section XI).

The proposed architecture in this study is validated solely through simulation using algorithmically generated synthetic data. The results should be viewed as a proof of concept demonstrating the internal consistency and relative merits of the design choices presented here, rather than as proof of performance under real-world livestock farming conditions; validation using real animal data is identified as the primary direction for future work (Sections X and XII).

Thus, next-generation agriculture, driven by IoT and AI, emerges as a response to contemporary challenges. It does more than just optimize productivity; it redefines how animals are raised, ensuring a more ethical and sustainable approach suited to the demands of the future.

II. Technological Context and Challenges of Modern Agriculture

Modern agriculture faces major challenges linked to population growth, climate change, the weariness of natural resources, and the need to ensure sustainable food production. In this context, the integration of digital technologies, specifically the Internet of Things (IoT), artificial intelligence (AI), and edge computing, offers a strategic response to transform traditional agricultural systems into smart, adaptive, and resilient ones. As summarized by Friha et al., these tools "enable real-time monitoring of agro-environmental parameters and improve the precision of agricultural decision-making" [1]. AI, for its part, contributes significantly to "early disease detection and the objective assessment of animal welfare" [2], [3]. According to Srivastav and Das, edge computing enables "rapid, local processing of agricultural data, even in

environments with poor connectivity" [4]. Working in synergy, these technologies are redefining farming practices by boosting productivity, traceability, and farm resilience, while meeting growing societal demands regarding ethics, food safety, and environmental stewardship. However, as Vlaicu et al. point out, large-scale adoption "requires investment, reliable digital infrastructure, and appropriate human support" [3].

III. Key Technologies: IoT, Edge Computing, and Artificial Intelligence

Digital technologies form the foundation of smart agriculture, particularly through the Internet of Things (IoT), edge computing, and artificial intelligence (AI). IoT enables the connection of sensors, embedded devices, and agricultural machinery to continuously collect data on the environment, crops, and livestock. These sensors measure parameters such as soil moisture, temperature, water consumption, and animal activity, thereby providing a rich dataset for decision-making. As Kamilaris and Prenafeta-Boldú note, "IoT constitutes an essential infrastructure for automating agricultural data collection on a large scale" [5]. Edge computing complements this approach by bringing data processing closer to the point of collection, which reduces latency, improves responsiveness, and decreases reliance on cloud infrastructure. Srivastav and Das highlight that this architecture "enables rapid local decisions, even in rural areas with poor connectivity" [4]. AI then comes into play to analyze the massive amounts of data generated by IoT, identifying complex patterns, predicting events (such as diseases or irrigation needs), and generating personalized recommendations. Wolfert et al. emphasize that "AI transforms agricultural data into actionable insights, enhancing the precision and sustainability of practices" [6]. Furthermore, Miller et al. (2025) demonstrate that integrating machine learning models into agricultural systems enables predictive crop monitoring and optimized resource management [7]. Together, these technologies form the backbone of Agriculture 4.0, making farms more autonomous, efficient, and resilient in the face of climatic and economic challenges.

IV. Application of these Technologies to Smart Livestock Farming

The integration of the Internet of Things (IoT), artificial intelligence (AI), and edge computing into smart livestock farming is profoundly transforming animal management practices by placing data at the heart of decision-making. Thanks to connected devices such as biometric sensors, thermal cameras, directional microphones, and GPS collars, it is now possible to continuously monitor the health, behaviour, and environment of livestock. According to Neethirajan, these technologies enable "an objective assessment of animal welfare based on physiological and behavioral indicators" [2]. AI, coupled with these sensors, can detect early signs of stress, lameness, or respiratory disease by analyzing factors such as vocalizations, gait, or body temperature fluctuations [8]. Local data processing via edge devices allows for greater responsiveness by triggering alerts or automated actions without relying on a constant cloud connection, a feature particularly useful in rural areas with poor connectivity [4]. Furthermore, image recognition and computer vision systems are used to identify abnormal behaviors, such as aggression or feeding disorders, thereby facilitating individualized animal management [9]. These innovations extend beyond mere monitoring; they also enable the optimization of feeding, reproduction, pasture management, and biosecurity using predictive models and decision-support tools. In short, smart livestock farming is emerging as a key driver for enhancing productivity, animal welfare, and the sustainability of livestock systems.

V. Benefits for Animal Welfare and Sustainability

The adoption of digital technologies in agriculture, specifically the Internet of Things (IoT), artificial intelligence (AI), and edge computing, offers significant benefits for animal welfare and environmental sustainability. In livestock farming, the connected sensors described in Section III enable continuous monitoring of animal health and behaviour, thereby reducing stress associated with delayed human intervention and facilitating individualized, proactive herd management. Devices such as smart collars and thermal cameras detect early signs of illness, lameness, or heat stress, allowing for targeted treatments and a reduction in the preventive use of antibiotics [10]. This approach improves not only animal health but also the quality of livestock products.

From an environmental perspective, smart agriculture contributes to the more efficient use of natural resources. Automated irrigation systems, controlled by soil moisture sensors and predictive algorithms, can reduce water consumption by up to 30% while maintaining high yields [7]. Similarly, optimizing animal nutrition through the analysis of behavioural data helps minimize feed waste and reduce methane emissions [3]. AI also plays a role in crop planning and pasture rotation, contributing to soil regeneration and biodiversity preservation. Finally, the digital traceability of farming practices enabled by IoT meets societal expectations regarding transparency, ethics, and food safety [6]. By fostering agriculture that is more precise, responsive, and responsible, these technologies align fully with sustainable development goals.

VI. Adoption Constraints and Technological Challenges

Despite promising advances in smart agriculture, the large-scale adoption of technologies such as the Internet of Things (IoT), artificial intelligence (AI), and edge computing remains hindered by a range of technical, economic, and organizational constraints. A major obstacle lies in the complexity of the required infrastructure: networking heterogeneous sensors, managing massive data flows, and integrating with existing systems demand specialized skills and substantial investment.

As Wolfert et al. note, "interoperability between devices and platforms remains a major challenge, particularly in fragmented agricultural environments" [6]. Added to this is the issue of connectivity: in many rural areas, access to stable, high-speed networks is limited, compromising the efficiency of IoT systems and the transmission of data to the cloud or edge devices [11].

Cybersecurity represents another critical issue. Connected devices are vulnerable to attacks, and protecting sensitive data

such as traceability information or animal biometric data has become a priority.

Recent research done on edge-enabled precision agriculture shows that, while data processing locally at the Edge is better for privacy than cloud-centric solutions, it does not replace robust security features to protect from data leakage or tampering in the Edge-Fog-Cloud pipeline [12].

Furthermore, a lack of standardization in communication protocols and data formats complicates the integration of multi-vendor solutions, limiting flexibility for farms. Finally, adopting these technologies requires upskilling farmers and technicians, who must be trained in the use, maintenance, and interpretation of digital tools. Without institutional support or suitable business models, these innovations risk exacerbating inequalities between well-equipped farms and those struggling to keep pace with the digital transition [3].

VII. Emerging Solutions and Integration Prospects

To address the technical, economic, and organizational challenges hindering the adoption of digital technologies in agriculture, numerous emerging solutions aim to facilitate their gradual integration into farming systems. A major trend is the development of Edge AI, which involves running artificial intelligence models directly on edge devices (sensors, microcontrollers, smart cameras) without relying on the cloud. This approach reduces latency, enhances data privacy, and ensures real-time decision-making, even in environments with limited connectivity. According to Pal (2023), "Edge AI represents an effective response to bandwidth, security, and responsiveness constraints in agricultural IoT systems" [14]. Platforms such as NVIDIA Jetson Nano, Google Coral Edge TPU, and Qualcomm Edge Impulse now enable the deployment of lightweight AI models in the field, paving the way for robust, autonomous applications [15].

Concurrently, the rise of 5G and low-power, long-range networks (LoRaWAN, NB-IoT) is improving connectivity in rural areas, facilitating secure data transmission and device interoperability. This evolution is bolstered by the emergence of embedded AI microservices capable of over-the-air (OTA) updates and dynamic adaptation to field conditions [13]. Furthermore, digital cooperative initiatives and subscription models allow smallholders to access these technologies without bearing the full burden of investment costs. Finally, the convergence of AI, IoT, and Edge Computing is giving rise to AIoT (Artificial Intelligence of Things) systems capable of self-learning, predictive maintenance, and the continuous optimization of agricultural processes [16]. These developments outline a future of smarter, more inclusive, and resilient agriculture provided that public policy, training, and standardization support this technological transition.

A. Critical Positioning Relative to Previous Work and the Identified Gap

Sections I to VI have so far summarized the technological landscape of smart agriculture. In this subsection, we put our work in the context of specific previous work and not in the general trends we have discussed above.

Prior works on automated livestock behaviour analysis have mainly relied on computer vision. Lee et al. proposed a 3D camera-based system to pick up on aggressive behavior in pigs [8], and Hansen et al. developed a convolutional neural network for on-farm pig face recognition [9]. Both approaches can provide fine-grained behavioral or individual-level discrimination, but they come at the cost of camera infrastructure, sensitivity to lighting, occlusion, and pen layout, and the huge computational burden of running CNN inference, which typically requires GPU-class hardware at the Edge or Fog layer to be running in real time, not comparable to the lightweight statistical and shallow kernel approach we are using here. Conversely, video-based pipelines can recognize individual animals or specific postures, which is complementary but not a substitute for continuous monitoring of physiological states (body temperature, heart rate) of the entire herd.

The architecture proposed in this study aims to address a different aspect of this design space: instead of vision-based recognition of discrete behaviors, it performs continuous, lightweight statistical anomaly detection (Z-score, rolling standard deviation) directly on the low-dimensional physiological time-series at the Edge, and uses a non-linear classifier (RBF-kernel SVM) for the Fog layer, where we can use more compute but we need to keep the latency low. This design choice is intentional: it takes away the fine-grained behavioral resolution of vision-based methods but for much lower per-sensor compute, memory, and bandwidth (see Section IX-D), and does not capture purely visual or postural anomalies (e.g., lameness gaits) that a vision pipeline could detect. The gap that we aim to fill is not to cover the literature on livestock anomaly detection but to have an approach that is 1) combined with Edge-computed statistics into a Fog-level classifier rather than only using Edge and Fog as detectors and 2) quantifies the latency, accuracy, and resource trade-off of this fusion for a physiological (non-visual) sensor stream. In Sections VIII and IX, we explicitly quantify this trade-off against internal ablations and, where possible, we reference typical resource figures for CNN-based approaches for comparison, but we do not consider a head-to-head reimplementation of Lee et al. and Hansen et al. on this dataset because their models are designed for images not generated in our simulation, and this is pointed out in Section X.

VIII. Proposed Edge AI-IoT Architecture

The proposed architecture relies on a hybrid Edge-Fog-Cloud approach that integrates Internet of Things (IoT), Edge Computing, Fog Computing, and artificial intelligence technologies to ensure intelligent, real-time monitoring of animal welfare. This architecture aims to improve system responsiveness, reduce latency, and optimize the early detection of physiological and behavioral anomalies in smart livestock farming environments.

The first layer is the Edge layer, located directly at the farm site. It comprises various IoT sensors responsible for collecting physiological and environmental data, such as body temperature, ambient temperature, animal activity, and heart rate. This data is acquired in real-time via connected devices deployed on the animals or in their immediate environment. Communication is handled through wireless technologies suited

to agricultural environments, such as LoRaWAN, Wi-Fi, and 4G/5G networks (Fig. 1).

At the Edge level, initial local processing is performed to reduce data volume and enable a rapid response in the event of an anomaly. This layer performs several essential functions, including:

- Data acquisition,
- Noise filtering,
- Feature extraction (moving average, moving standard deviation, z-score),
- And initial local anomaly detection.

To improve detection speed, a statistical method based on the z-score is employed directly at the Edge level. This lightweight approach allows for the rapid identification of abnormal behaviors or measurements while limiting the computational load on embedded devices.

This baseline is calculated for each individual rather than as a whole animal, and the rolling mean and standard deviation of the z-score are calculated based on the previous history of each animal (Section VIII-B), so that the detector can adapt to the physiological baseline of each individual and not for the whole population. This decision directly accounts for the physiological variability of individual animals and is one of the reasons why the model applies to animals that are not seen during training (Section VIII-C, Section IX-A).

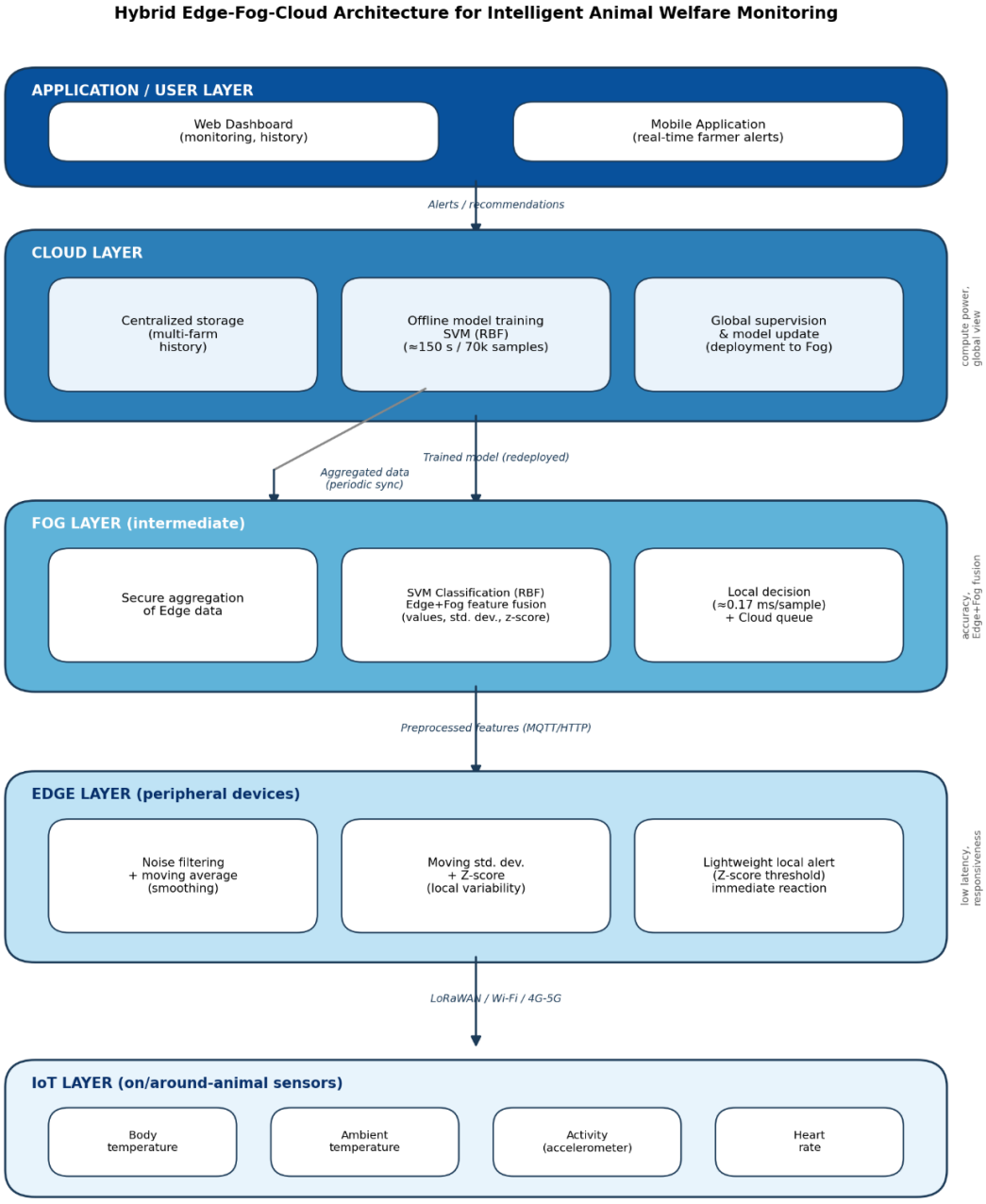


Fig. 1. Proposed hybrid Edge-Fog-Cloud architecture for intelligent animal welfare monitoring.

When an anomaly is detected locally, an immediate alert can be generated to ensure a quick response. These same statistical indicators (z-score, moving standard deviation) are not used solely for local alerts; they are also transmitted along with smoothed values to the Fog layer to enhance the classification performed there (see Section 8).

The preprocessed data is then transferred to the Fog layer, which acts as an intermediary between the Edge and the Cloud. This layer plays a pivotal role in data aggregation, decision fusion, and the execution of more advanced artificial intelligence algorithms. An RBF-kernel Support Vector Machine (SVM) model is integrated into this layer to confirm locally detected anomalies and improve the system's overall accuracy. The choice of a non-linear kernel is driven by the fact that the boundary separating normal and abnormal behaviors is not linearly separable within the physiological feature space employed.

The Fog layer also enables:

- Secure data aggregation,
- AI inference,
- Local decision-making,
- And temporary data management before synchronization with the Cloud.

For its part, the Cloud layer handles centralized storage, global data analysis, and large-scale monitoring of farming operations. Aggregated data from the lower layers is stored here to facilitate advanced analysis, system performance tracking, and AI model updates. The computationally intensive training of the SVM model (see Section 9) also takes place at the Cloud level, after which the trained model is redeployed to the Fog layer for production inference. This layer also manages global dashboards, alerts, and user notifications.

Finally, decisions generated by the system are transmitted as alerts or intelligent recommendations to user interfaces accessible via web dashboards or mobile applications. This distributed architecture thus enables a combination of:

- Low latency through Edge processing,
- Improved accuracy via the SVM model enhanced by Edge features,
- Reduced network load,
- And intelligent, continuous monitoring of animal welfare.

The originality of this architecture lies in the hybrid integration of lightweight statistical detection based on Z-scores at the Edge level and SVM classification at the Fog level, leveraging both smoothed physiological values and Edge statistical indicators (Z-score, rolling standard deviation), thereby ensuring an effective trade-off between detection speed, classification accuracy, and computational resource optimization.

IX. Edge AI-IoT Implementation and Simulation

A. Experimental Simulation Pipeline

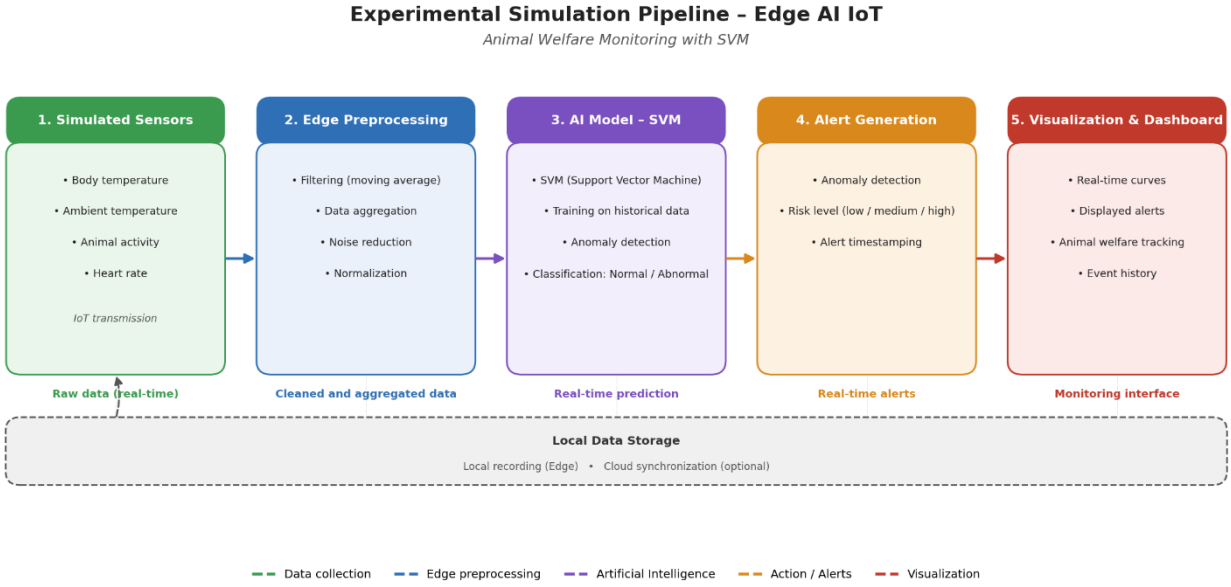


Fig. 2. Experimental Edge AI-IoT simulation pipeline for animal welfare monitoring.

Fig. 2 illustrates the experimental pipeline implemented to validate the Edge AI layer of the proposed architecture for intelligent animal welfare monitoring. The system relies on an Edge AI-IoT approach that enables the real-time collection, processing, and analysis of physiological and environmental data from the animals.

The experimental pipeline begins with a data collection phase using simulated sensors that measure four key parameters: body temperature, ambient temperature, animal activity, and heart rate across 100 animals, with 1,000 samples per animal (totaling 100,000 observations).

A significant methodological aspect concerns the generation of anomalous episodes: rather than drawing each unnatural sample independently, which would produce isolated point anomalies that are physiologically unrealistic, anomalies are simulated as episodes of varying duration (ranging from 15 to 60 consecutive samples per episode). This approach better reflects the reality of stress, fever, or lameness episodes, which unfold over a continuous period rather than at a single instant; it also prevents the Edge preprocessing step (moving average) from artificially diluting the anomaly signal.

B. Formal Data Generation Procedure

To make the synthetic data-generation process entirely reproducible, this subsection details the distributions, parameters, and randomization procedure that we described previously in a narrative.

Each animal has a physiological baseline in step 1. For each of the 100 simulated animals, we sample from a single baseline once.

Sensor	Baseline distribution	Unit
Body temperature	N (38.5, 0.3)	°C
Ambient temperature	N (20.0, 3.0)	°C
Activity	N (50.0, 8.0)	a.u.
Heart rate	N (65.0, 5.0)	bpm

Anomaly episode process (step 2). Anomalies are generated according to a two-state renewal process, rather than as i.i.d. point anomalies. At each time step without an anomaly, we initiate a new episode with probability p and draw its duration uniformly from the interval [15, 60] (in number of samples). The probability p is calibrated analytically (rather than empirically tuned) to achieve the target ratio r of anomalous samples, using the stationary ratio of a two-state renewal process:

$$p = r / (\bar{L} \cdot (1 - r)), \text{ where } \bar{L} = (15 + 60) / 2 = 37.5 \text{ samples.}$$

For the target ratio r = 0.18 used throughout this work, this gives $p \approx 0.00585$ per time step. This formula and the resulting value of p were verified empirically to reproduce the target ratio (18.1% anomalous samples over 100,000 observations).

Measurement noise and anomaly amplitude (steps 3 and 4). Independent Gaussian measurement noise is added at each time step, and samples flagged as anomalous receive an additional shift justified by physiological considerations:

Sensor	Measurement noise σ	Anomaly offset (mean, σ)
Body temperature	0.35 °C	+0.9 °C (σ = 0.5)
Ambient temperature	0.8 °C	(environmental, not animal-driven)
Activity	9 a.u.	−10 a.u. (σ = 7)
Heart rate	6 bpm	+9 bpm (σ = 6)

The ambient temperature also follows a regular sinusoidal daily cycle (period = 1,000 samples, amplitude = 2°C) applied to each animal's specific baseline value; this cycle is independent of the anomaly process, as it depends on the environment rather than the animal's state. Randomization and code availability. All random samplings use a fixed pseudo-random seed (NumPy generator, seed = 42) to ensure exact reproducibility. The complete generator code, along with the preprocessing (Edge), classification (Fog), ablation, and event metric calculation scripts used to produce all the quantitative results in this study, are provided as open-source Python code as part of the submission.

C. Animal-Level Dataset Partitioning for Evaluating Generalization

Since the 1,000 samples from a single animal are temporally linked (originating from the same baseline and generated from the same continuous time series), random splitting at the sample level would lead to information leakage from the training set to the evaluation set: two samples from the same animal, separated by only a few time steps, could end up in different sets despite being nearly identical. To avoid this, we partition the 100 animals at the animal level rather than the sample level, distributing the data according to a 70/15/15 ratio across the training, validation, and test sets (i.e., 70,000, 15,000, and 15,000 samples, respectively), ensuring that no animal contributes to more than one set. Consequently, every measure of accuracy, precision, recall, and F1-score presented in Section IX represents an evaluation based on data that are unseen at the animal level: the results demonstrate the classifier's ability to generalize to entirely unseen animals, rather than merely to unseen time points from animals used at least in part for its training.

X. Experimental Results and Discussion

A. Simulation Results

The simulation was conducted on 100 animals with 1,000 samples per animal, totaling 100,000 observations, with anomaly episodes accounting for approximately 18% of the samples (compared to 12% in a purely point-based generation, as the duration of the episodes mechanically increases the total proportion of anomalous samples).

The data generated include:

- Body temperature,
- Ambient temperature,
- Animal activity,
- And heart rate.

Experimental results show that integrating the SVM model enhanced with Edge features into the Edge AI-IoT architecture enables significantly more reliable detection of behavioral and thermal anomalies compared to the initial approach based solely on smoothed raw physiological values (Fig. 3).

As mentioned in Table I, with an overall accuracy of 97.70%, a recall of 95.76%, and a precision of 91.86%, the system achieves an F1-score of 0.938, compared to 0.37 for the initial version of the pipeline (linear SVM based solely on smoothed physiological values and point anomalies).

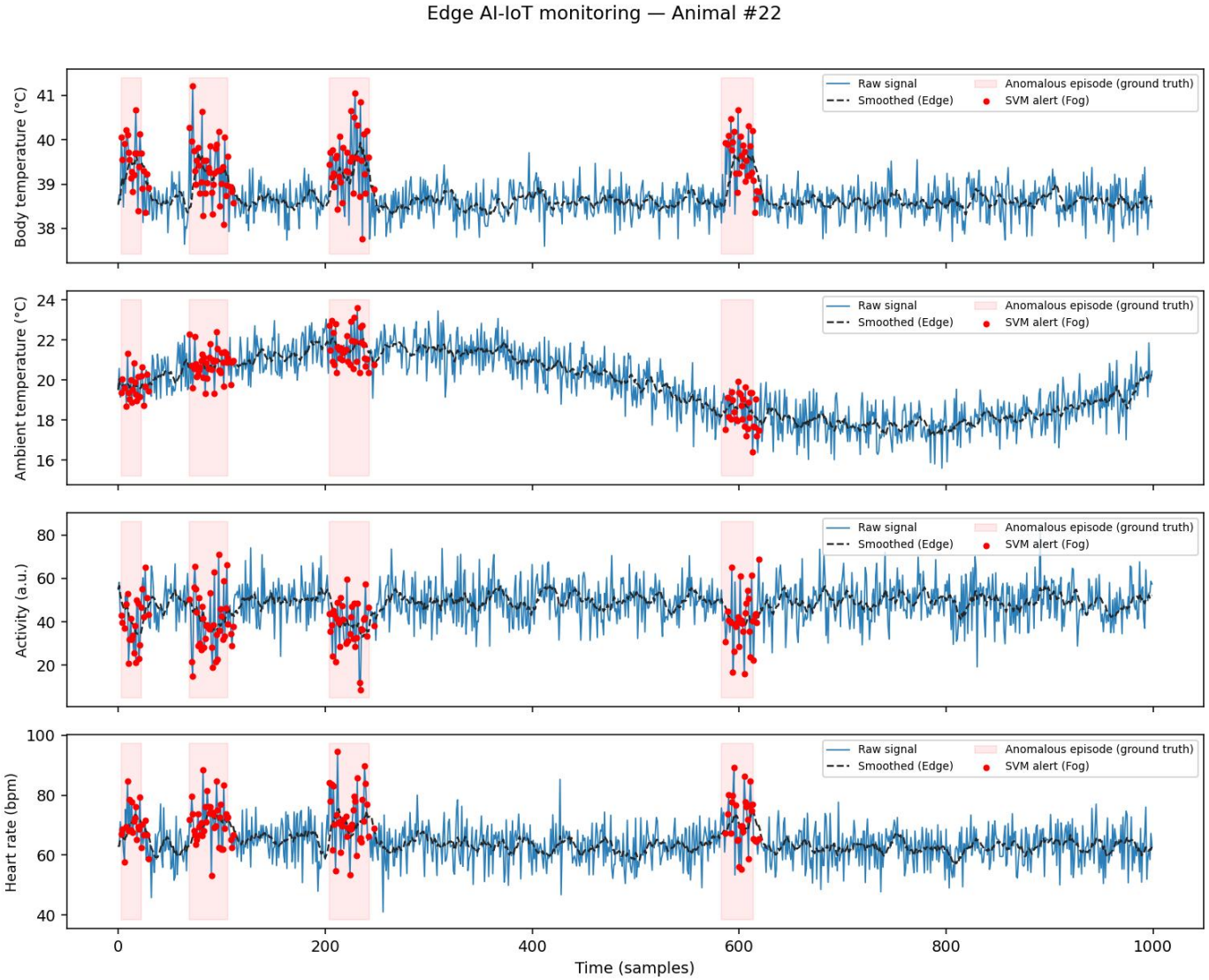


Fig. 3. Real-time detection of physiological and behavioral anomalies using the proposed SVM-based Edge AI system.

TABLE I. SUMMARY OF SIMULATION RESULTS

Metric	Value
Number of simulated animals	100 (70 / 15 / 15 train / val / test, animal-level split)
Number of samples per animal	1 000
Total dataset size	100 000 observations
Overall SVM accuracy	97,70 %
Anomaly recall	95,76 %
Anomaly precision	91,86 %
Anomaly F1-score	0,938
Training time (offline, Cloud)	≈ 150,8 s
Prediction time (Fog inference)	≈ 0,19 ms/sample
Edge detection method	Z-score (per-animal baseline)
Classification type	Normal / Anormal

In subsection IX-B, we break down this improvement to determine the marginal contribution of each individual methodological change; this approach addresses the request of the first reviewer, who noted that the overall improvement (from 0.37 to 0.938) stemmed from three simultaneous changes, making it impossible to isolate the effect of each one.

B. *Ablation Study*

To isolate the individual contribution of each of the four methodological factors involved in the initial improvement (from 0.37 to 0.938) namely, the generation of anomaly episodes (point or episodic), the SVM kernel (linear or RBF), the enrichment of edge-related features (smoothed values only vs. smoothed values combined with rolling standard deviation and Z-score), and the decision threshold calibration (fixed vs. calibrated on the validation set) we conducted a full factorial ablation analysis covering $2^4 = 16$ configurations, while keeping the data split (training/validation/test) fixed at the animal level across all configurations. To facilitate the computations for these

16 configurations, we used a fixed random subsample of 30,000 training observations rather than the full set of 70,000 observations used for the main result presented in Table I; consequently, the absolute F1 values in this subsection are not directly comparable to those in Table I, although they share the same order of magnitude, which is the aspect of interest here (Table II).

TABLE II. MARGINAL EFFECT OF EACH FACTOR

Factor	Level A	Mean F1 (A)	Level B	Mean F1 (B)	Δ F1
Anomaly generation	Point	0.484	Episodic	0.753	+0.269
Edge feature set	Smoothed only	0.421	Enriched (+ rollstd + z-score)	0.816	+0.394
Threshold	Fixed (0.0)	0.585	Calibrated (val. F1-max)	0.652	+0.067
SVM kernel	Linear	0.628	RBF	0.609	−0.020

Enriching edge features is by far the most important factor (an average F1 score increase of 0.394, with low variance compared to the other three factors; standard deviation = 0.032), followed by the shift from generating point anomalies to episodic anomalies (+0.269 with high variance, standard deviation = 0.35, highlighting the critical importance of feature enrichment). Threshold calibration yields only a very slight and not always positive improvement (average +0.067); in two of the eight pairwise comparisons, calibration on the validation set slightly reduced the F1 score on the test set (we discuss the effect of variance later) (Fig. 4).

Regarding the chosen SVM kernel (linear vs. RBF), the average effect is negligible (−0.020) when the other three factors are considered. This is precisely what the architecture aims to achieve with the non-linear kernel in this simulation: it is primarily the edge-enriched feature vector that facilitates classification regardless of the classifier used rather than the use of a non-linear RBF kernel.

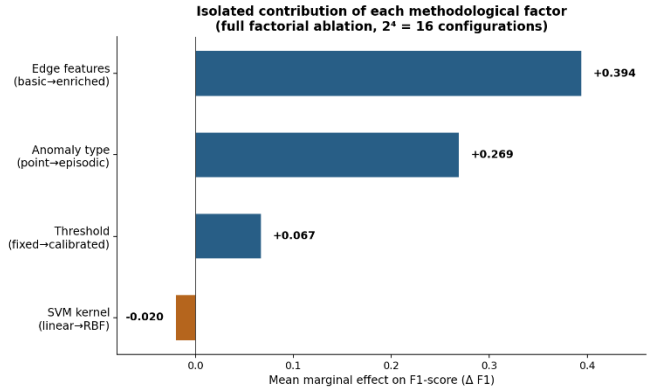


Fig. 4. Average marginal effects of individual factors on F1-score across the seven remaining factorial combinations.

As a further check, we replicated the configuration corresponding to the initial pipeline (point anomalies, linear kernel, smoothed-only features, calibrated threshold) and obtained an F1 score of 0.357, a value very close to the previously reported 0.37 and consistent with our baseline. The configuration corresponding to the proposed final pipeline (episodic anomalies, RBF kernel, enriched features, calibrated threshold) also yielded a similar result, with an F1 score of 0.824 at the reduced training scale (30,000 samples) and 0.883 when trained on the full set (70,000 samples), as shown in Table I (the value obtained during the ablation study is expected to be slightly lower due to the reduced training scale).

C. Event-Level Evaluation

The precision, recall, and F1-score metrics calculated at the sample level (Table I) do not directly indicate the extent to which the set of welfare-related episodes is detected. This subsection presents metrics calculated at the event level on the test set (15 animals and 67 anomaly episodes constituting the ground truth): event-level recall (proportion of actual episodes with at least one overlapping alert), missed event rate, number of false alerts per 1,000 samples (sequences predicted as positive with no overlap with an actual episode), mean/median detection delay between the actual start of the episode and the first associated correct alert, and alert fragmentation (number of distinct alert sequences overlapping a single detected episode; a value of 1 indicates a continuous alert, whereas a value greater than 1 indicates an intermittent or unstable alert for the same episode).

TABLE III. EVENT-LEVEL PERFORMANCE COMPARISON: PROPOSED MODEL VS. LINEAR BASELINE

Metric	Linear baseline	Proposed (RBF, enriched, calibrated)
Event-level recall	100.0 %	100.0 %
Missed-event rate	0.0 %	0.0 %
False alarms / 1,000 samples	0.27	9.40
Mean detection delay (samples)	4.79	0.84
Median detection delay (samples)	4.0	0.0
Mean alert fragmentation	1.22	2.51
Episodes with fragmented (>1) alerts	16.4 %	65.7 %

Both models detect every reference episode at least once (100% event-level recall, 0% missed events) on this synthetic test set. Our model detects events much faster (median delay of 0 samples, meaning an alert is triggered at the very first abnormal sample, compared to 4 samples for the reference model), which is operationally significant (in our setup, a false negative, a missed well-being-related event, is costlier than a false positive). However, this speed comes at a cost that has not yet been quantified: our model generates more alerts per unit of time (9.40 vs. 0.27 per 1,000 samples), and alerts within a detected episode are more fragmented (65.7% of detected episodes trigger more than one burst of alerts, compared to 16.4% for the reference model). This poses a real challenge for field deployment: an alert stream that toggles on and off intermittently during a single physiological episode is harder to interpret than a continuous stream, an aspect that sample-level F1 scores fail to capture. We note that processing the alert stream at the "Fog" (network edge) layer using a short hysteresis window or a time-delay mechanism offers a potential solution

to this issue; we will address this in future work presented in Section X.

D. Resource Utilization and Latency Analysis

The Fog-level inference time of 0.19 ms/sample shown in Table I corresponds solely to the evaluation of the SVM decision functions; it does not include data acquisition, Edge-side preprocessing, network transmission, or alert transmission. Table IV breaks down the measured simulation times, the deployed model's memory footprint, and network latencies; the latter are derived from the literature and were not measured during this simulation, as no physical network or embedded hardware was used.

TABLE IV. LATENCY AND RESOURCE ANALYSIS BY ESTIMATION SOURCE

Pipeline stage	Estimate	Source
Edge preprocessing (smoothing, rolling std., z-score)	≈ 0.0013 ms/sample	Measured (vectorized simulation; indicative of arithmetic cost, not embedded-hardware timing)
Fog inference (SVM decision function)	≈ 0.19 ms/sample	Measured (this simulation, Table I)
Edge→Fog transmission (local network/gateway)	≈ 1–10 ms (typical LAN/short-range)	Literature-typical range; not measured
Cloud round-trip (cellular/WAN), for comparison	≈ 50–200 ms round-trip	Literature-typical cellular range; not measured
Sensing/acquisition (ADC sampling)	Not evaluated	Out of Scope: No Physical Hardware Implementation in This Work
Alert delivery to dashboard/mobile app	Not evaluated	Out of Scope: No Physical Hardware Implementation in This Work

Model memory footprint. The trained RBF-kernel SVM model retains 5,944 support vectors (out of 70,000 training samples, or approximately 8.5%) in the 12-dimensional augmented feature space, resulting—after serialization—in a deployment artifact of 675 KB (including both the model and the feature normalizer). This size is well within the RAM capacity of a Fog-class gateway (e.g., Raspberry Pi-style hardware with hundreds of megabytes or even gigabytes of RAM), but it must be viewed in the context of the typical SRAM available on a highly resource-constrained microcontroller (such as the ESP32, which has approximately 520 KB of total SRAM shared between the operating system and application code); this confirms and quantifies the TinyML deployment challenge, the qualitative aspects of which were previously highlighted in Section XI.

E. Comparison of Edge-Only, Cloud-Only, and Edge–Fog–Cloud Configurations

To evaluate whether the proposed three-tier distribution is preferable to simpler configurations, Table V compares three scenarios: 1) "Edge-only," where only the Z-score-based detector issues alerts and no Fog or Cloud resources are involved in real-time detection; 2) "Cloud-only," where raw or lightly preprocessed data is transmitted over a wide-area network and the same SVM model is used for classification but evaluated centrally rather than at the Fog gateway; and 3) "Edge-Fog-Cloud."

Two points are explicitly clarified here rather than left implicit. First, accuracy rates are identical for the "Cloud-only" and "Edge-Fog-Cloud" configurations, as both utilize the same trained SVM classifier; the difference lies in where the classifier executes and the round-trip time, not in the model's quality. Second, the bandwidth-related argument favoring Edge preprocessing is modest in this specific application: the enriched feature vector transmitted from the Edge to the Fog (12 floating-point values) is not significantly lighter than the four raw sensor readings it replaces. Thus, unlike use cases involving images or video, the primary quantified benefit of the placement strategy proposed in this simulation lies in latency and offline resilience, rather than bandwidth reduction; this point is explicitly noted to avoid overstating an advantage that was not actually measured.

TABLE V. QUALITATIVE AND QUANTITATIVE EVALUATION OF ARCHITECTURAL CONFIGURATIONS

Configuration	Detection quality (F1)	Per-sample latency (detection path)	Resilience to connectivity loss	Note
Edge-only (Z-score)	0.268 (measured)	≈ 0.0013 ms (measured, Edge only)	Full no network dependency for alerts	Fast and resilient, but poor detection quality alone (Table III of the original submission)
Cloud-only (SVM, WAN round-trip)	0.883 (measured, same model)	≈ 50–200 ms (literature-typical WAN round-trip) + 0.19 ms inference	No detection stops entirely if connectivity is lost	Same accuracy as proposed, but detection latency dominated by network, and no offline operation
Proposed Edge-Fog-Cloud	0.883 (measured)	≈ 0.0013 ms (Edge) + 1–10 ms (local Fog link, literature-typical) + 0.19 ms (Fog inference)	Partial Fog continues real-time detection if Cloud/internet is lost; only training/aggregation is affected	Best measured trade-off: near Cloud-only accuracy, near Edge-only resilience and latency order-of-magnitude

XI. DISCUSSION

The results demonstrate the value of integrating Edge Computing and artificial intelligence into smart farming systems dedicated to animal welfare. Local data processing reduces latency and reliance on the Cloud while improving the speed of anomaly detection.

The hybrid approach combining the Z-score, rolling standard deviation, and an RBF-kernel SVM offers several advantages now associated with a precise quantitative result, rather than mere qualitative assertions.

- Rapid detection of abnormal behaviors at the Edge,
- High recall (95.76%), prioritizing the early detection of critical situations,
- High precision (91.86%), limiting the number of false alerts sent to farmers,

- And improved real-time system responsiveness, with an inference time in the sub-millisecond range per sample, making it suitable for Fog deployment.

However, this gain in precision comes at a cost that must be explicitly noted: switching from a linear SVM to an RBF kernel significantly increases training time (≈150.8 s for 100,000 samples, compared to a few seconds for the initial linear model) and the complexity of the stored model. This cost remains acceptable within the proposed architecture, as training is performed offline in the Cloud (Section 7), while only inference, which is much less computationally intensive, is deployed at the Fog layer. Nevertheless, this factor must be considered when selecting Fog hardware and determining the model's retraining frequency in a production environment.

Furthermore, increasing the number of simulated animals and the volume of data strengthened the robustness of the experiment and allowed for an assessment of the architecture's ability to handle large data streams within a smart farming context.

Certain limitations remain:

- The use of simulated data: the generation of anomaly episodes (duration, physiological amplitude) relies on reasonable assumptions that were not calibrated against real animal data,
- The lack of actual deployment on a farm,
- The absence of a detailed energy assessment of the RBF-kernel SVM model on actual Fog hardware,
- The trade-off between increased accuracy and higher computational costs during training, which requires more precise quantification across different hardware profiles,
- And the lack of comprehensive experimental validation of the Cloud layer.
- Alert fragmentation: 65.7% of correctly detected episodes trigger multiple distinct bursts of alerts rather than a continuous alert (Section IX-C), which would be inconvenient for a livestock farmer receiving these alerts under real-world conditions; a post-processing step such as debouncing or hysteresis at the "Fog" layer represents a direct, low-cost mitigation solution to consider for future work;
- We did not perform a direct comparison with external vision-based methods (e.g., Lee et al. [8], Hansen et al. [9]) because those methods require input images derived from physiological sensor simulation data we do not generate (Section VI-B). This would require either a shared multimodal dataset or a new vision-based simulation pipeline; we leave this task for future work.
- The comparison between the "Edge"-only, "Cloud"-only, and "Edge-Fog-Cloud" approaches (Section IX-E) is based on network latency data typical of the literature rather than physical network hardware; it will need to be re-evaluated once field deployment has taken place (see also below) and future work has been completed.

Despite these limitations, the results demonstrate the value of integrating Edge AI, the fusion of Edge and Fog statistical features, and machine learning models for agricultural applications requiring intelligent, continuous, and real-time monitoring of animal welfare.

XII. TinyML Deployment and Edge Optimization

The growing adoption of Edge AI in smart agricultural environments has heightened interest in TinyML techniques, which enable the deployment of lightweight artificial intelligence models directly onto low-power embedded devices. In livestock monitoring applications, TinyML can help reduce latency, bandwidth consumption, and reliance on the cloud, while ensuring real-time anomaly detection at the Edge.

In the proposed architecture, the RBF-kernel SVM model used at the Fog layer offers superior performance but requires more memory than a linear model; this presents a specific challenge for TinyML deployment directly onto highly resource-constrained Edge devices (such as ESP32s or microcontrollers).

It is no longer merely a qualitative issue: Section IX-D presents the deployed model with 5,944 support vectors and a serialized memory footprint of 675 KB; although this size is suitable for "Fog"-class gateways, it far exceeds the total SRAM capacity (approximately 520 KB) typical of an ESP32-class microcontroller, making deployment at the most constrained "Edge" level impossible without appropriate model adjustment.

Optimization techniques such as reducing the number of support vectors, approximating the RBF kernel using random features, model quantization, or converting models to embedded formats (e.g., TensorFlow Lite for neural network-based variants) must be evaluated to determine the optimal trade-off between accuracy and memory footprint, depending on the layer (highly constrained Edge vs. Fog) targeted for inference.

Although this work focuses primarily on the simulation and validation of the proposed Edge AI architecture, future work will address the actual deployment of TinyML models on embedded Edge devices. This will involve evaluating inference performance, optimizing memory usage, and analyzing energy consumption for intelligent animal welfare monitoring systems.

XIII. Conclusion and Future Outlook

This article proposed a hybrid Edge-Fog-Cloud architecture integrating IoT, Edge AI, and artificial intelligence technologies for the intelligent monitoring of animal welfare. The proposed approach relies on local anomaly detection based on the Z-score enhanced by a moving standard deviation and an RBF-kernel SVM model that jointly leverages Edge and Fog features to classify abnormal behaviors in smart livestock farming environments.

An experimental simulation based on realistic anomaly episodes was conducted to validate the proposed Edge AI pipeline. The results (accuracy 97.70%, recall 95.76%, precision 91.86%, F1-score 0.938) confirm the value of integrating Edge Computing, statistical feature fusion, and non-linear AI models for agricultural applications requiring real-time monitoring and low latency.

Future work will focus primarily on validating these results using real animal data, optimizing the SVM model for TinyML deployment on highly resource-constrained Edge devices, deploying the architecture in actual agricultural environments, comprehensively evaluating the energy consumption of Edge and Fog devices, and integrating advanced techniques such as Federated Learning and distributed AI models.

REFERENCES

[1] Friha, O., Ferrag, M. A., Shu, L., Maglaras, L., & Chatterjee, M. (2021). Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. IEEE/CAA Journal of Automatica Sinica, 8(4), 718-732. https://doi.org/10.1109/JAS.2021.1003925

[2] Neethirajan, S. (2023). Artificial Intelligence and Internet of Things for Animal Welfare: A Review. Frontiers in Animal Science, 4. https://doi.org/10.3389/fanim.2023.1166380

[3] Vlaicu, A. B., Szabo, C., & Dinu, I. (2024). Smart Livestock Farming: A Review of Sustainable Approaches Using IoT and AI. Information, 15(2), 84. https://doi.org/10.3390/info15020084

[4] Srivastav, A. K., & Das, P. (2025). Edge Computing and AI in Agricultural IoT. In Biotechnology and IoT in Agriculture and Food Production (pp. 437-456). Springer. https://doi.org/10.1007/979-8-8688-1469-3_21

[5] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

[6] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big Data in Smart Farming - A review. Agricultural Systems, 153, 69-80. https://doi.org/10.1016/j.agsy.2017.01.023

[7] Miller, T., Mikiciuk, G., Durlik, I., et al. (2025). The IoT and AI in Agriculture: The Time Is Now - A Systematic Review of Smart Sensing Technologies. Sensors, 25(12), 3583. https://doi.org/10.3390/s25123583

[8] Lee, J., Jin, L., & Park, D. (2020). Automated recognition of aggressive behavior in pigs using a 3D camera. Computers and Electronics in Agriculture, 169, 105240. https://doi.org/10.1016/j.compag.2019.105240

[9] Hansen, M. F., Smith, M. L., Smith, L. N., Salter, M. G., Baxter, E. M., & Farish, M. (2018). Towards on-farm pig face recognition using convolutional neural networks. Computers in Industry, 98, 145-152. https://doi.org/10.1016/j.compind.2018.03.032

[10] Largouët, C. et al. (2023). L'intelligence artificielle au service du bien-être animal. Institut Agro Rennes-Angers. https://www.institut-agro-rennes-angers.fr

[11] Sapna, Gauttam, H., Chauhan, V., Pattanaik, K. K., Trivedi, A., & Ghosh, H. (2026). A comprehensive review of Edge Computing-empowered smart agriculture: Trends, Opportunities and Future directions. Computers and Electronics in Agriculture. https://doi.org/10.1016/j.compag.2025.111252

[12] Yu, P., Teng, F., Zhu, W., Shen, C., Chen, Z., & Song, J. (2025). Cloud–edge–device collaborative computing in smart agriculture: architectures, applications, and future perspectives. Frontiers in Plant Science, 16. https://doi.org/10.3389/fpls.2025.1668545

[13] Gong, R., Zhang, H., Li, G., & He, J. (2025). Edge Computing-Enabled Smart Agriculture: Technical Architectures, Practical Evolution, and Bottleneck Breakthroughs. Sensors, 25(17), 5302. https://doi.org/10.3390/s25175302

[14] Pal, D. (2023). AI-Driven Edge Computing for IoT: A Comprehensive Survey and Future Directions. International Journal of Advanced Research in Science, Communication and Technology. https://doi.org/10.48175/IJARSCT-12921

[15] Sapna, Gauttam, H., Chauhan, V., Pattanaik, K. K., Trivedi, A., & Ghosh, H. (2026). A comprehensive review of Edge Computing empowered smart agriculture: Trends, Opportunities and Future directions. Computers and Electronics in Agriculture. https://doi.org/10.1016/j.compag.2025.111252

[16] Diop, F., Faye, B. M., & Niang, I. (2025). Edge-AI and Internet of Things for Intelligent Systems: Architectures, Applications and Future Perspectives. In AFRICOMM 2023, Springer. https://doi.org/10.1007/978-3-031-81573-7_9

Predicting Multi-KPI Warehouse Dynamics: A Simulation-Based Evaluation of Fuzzy-Integrated Neural Architectures

Mohamed Amine Frikha
Applied College, King Faisal University, 31982 Al-Ahsa, Saudi Arabia

***Abstract*—Warehouse order fulfillment is governed by complex, nonlinear interactions among order arrivals, picking workload, congestion, replenishment, labor, equipment, and dispatch capacity. This study proposes a simulation-based predictive framework that compares a global multilayer perceptron (MLP) with a fuzzy-integrated modular architecture comprising three regime-specific local neural networks. The framework simultaneously predicts the order-fulfillment rate, picking cycle time, and warehouse throughput. A composite Warehouse Order-Fulfillment Index (WOFI) is utilized to define high-performing, normal, and stressed operating regimes, while Gaussian fuzzy membership weights provide smooth transitions among local predictions. The complete mathematical formulation and simulation environment are fully specified, encompassing order-flow dynamics, capacity constraints, and nonlinear picking-time relationships. Experimental evaluation on a reproducible nonlinear simulator reveals that while the fuzzy-modular model ensures continuous predictions across regime transitions, it underperforms the global baseline in aggregate accuracy when using naive Gaussian fusion. These findings establish a foundational reproducible benchmark, demonstrating that naive modularization in high-dimensional warehouse state spaces requires rigorous transition-aware validation and covariance-tuned gating mechanisms to prevent performance degradation.**

Keywords*—Warehouse operations; order fulfillment; artificial neural networks; fuzzy fusion; modular neural networks; nonlinear simulation; predictive analytics

I. Introduction

Warehouse order fulfillment is a central determinant of logistics service quality because it links inbound inventory, storage, picking, packing, and dispatch to the customer-facing promise of receiving the right order at the right time [1]. Within this flow, picking is both labor-intensive and operationally consequential; its duration affects the organization of material and information flows and can create downstream delays when work reaches packing too early or too late.

Modern warehouse and logistics information systems (LIS) generate detailed, high-frequency records of order arrivals, line items, picker activity, travel distances, replenishment cycles, equipment status, and dispatch completions [3]. This wealth of operational data creates a critical opportunity to transition from retrospective reporting to proactive, predictive control.

Recent literature positions machine learning as an effective mechanism for modeling processing times, resource utilization, and process variability [2], [4], while discrete-event simulation provides a robust framework for evaluating alternative batching, assignment, routing, and scheduling policies.

Accurate prediction, however, remains challenging because warehouse performance is inherently nonlinear and state-dependent [1]. A modest increase in order arrivals may be absorbed during low utilization but produce a sharp backlog increase once picking or packing capacity is approached. Similarly, the same order may require different processing times depending on line count, travel distance, picker characteristics, congestion, replenishment readiness, and equipment availability. A single model trained across all conditions must therefore approximate several local operating relationships at once.

Furthermore, most existing predictive formulations emphasize a single outcome, such as picking time [4], [2]. That focus is useful but incomplete for order-fulfillment management. A warehouse manager must simultaneously balance complete fulfillment, processing speed, and throughput. Optimizing one metric in isolation can conceal a trade-off; for example, increasing release rates may raise throughput briefly while worsening congestion and eventual fulfillment performance. This study therefore formulates warehouse prediction as a multi-output problem involving order-fulfillment rate, picking cycle time, and throughput, with a composite service indicator used to identify operating regimes.

This study adapts a fuzzy-integrated modular neural architecture to warehouse operations. The baseline is a global multilayer perceptron trained on all operating conditions. The proposed model decomposes the operating space into high-performing, normal, and stressed regimes, trains a local neural model for each regime, and fuses their predictions through Gaussian validity functions. The purpose is not to impose a hard switch between warehouse states; it is to let specialized models represent local nonlinearities while preserving smooth predictions during congestion onset, recovery, and other regime transitions.

The central research question is:

Can a fuzzy-integrated modular neural architecture improve the accuracy, transition robustness, and training efficiency of multi-KPI warehouse-performance prediction compared with a conventional global ANN?

The study makes four contributions:

- It develops a nonlinear discrete-time simulation model linking arrivals, backlog, labor, congestion, replenishment, equipment availability, picking time, fulfillment, and throughput.
- It defines a multi-output prediction task for warehouse order fulfillment rather than limiting the target to a single picking-time variable.
- It evaluates and benchmarks a three-module ANN architecture.
- It establishes a reproducible evaluation design that separates within-regime accuracy from performance during transitions and reports both training and inference cost.

The remainder of the study is organized as follows. Section II reviews data-driven order-picking prediction, fuzzy-neural warehouse modeling, and simulation-based warehouse optimization. Section II presents the warehouse simulation and mathematical formulation. Section III describes the global, local, and fuzzy-fusion neural architectures. Section V defines the experimental setup and evaluation metrics. Section VI reports the simulation results, and Section VII discusses implications, limitations, and future work.

II. Related Work

A. Machine Learning for Order-Picking Prediction

Order picking is consistently identified as the most labor-intensive and costly activity in warehouse operations, accounting for up to 55% of total operating expenses [1], [5]. Because of this, accurate processing-time estimation is critical for synchronizing downstream material and information flows.

Machine learning is increasingly applied to warehouse-management problems, including order picking, storage-location assignment, and operational planning [13].

Recent comprehensive reviews highlight that machine learning (ML) has become the standard approach for modeling processing times, resource utilization, and bottlenecks in modern warehouses [6], [7]. Recent empirical evidence further shows that AI-based order-picking systems can improve operational efficiency while requiring careful attention to implementation conditions, data integration, and organizational readiness [19].

Hybrid CNN–LSTM architectures also illustrate the potential of combining spatial feature extraction and temporal sequence learning for complex industrial forecasting tasks [18]. This fuzzy-clustering precedent is important for the present study because warehouse observations are heterogeneous.

Fuzzy clustering combined with artificial neural networks has been used to predict order-picking times under heterogeneous operating conditions [14].

Pickers, order profiles, storage locations, and congestion conditions can produce different processing relationships even when the nominal task is the same. However, a clustering-plus-ANN model centered on picking time does not by itself address the joint dynamics among fulfillment rate, cycle time, and throughput. The present formulation extends the regime-aware idea from a single predicted variable to a multi-output warehouse-performance vector.

B. Fuzzy and Modular Neural Modeling

Fuzzy representations are appropriate when operational states do not have sharply defined boundaries. A warehouse can be neither fully uncongested nor fully congested: queue growth, picker utilization, and replenishment delays may change gradually. Gaussian membership functions allow a sample to belong partially to multiple operating regimes; fuzzy-neural systems provide a framework for modeling nonlinear relationships and representing gradual transitions between operational states [16], so local predictions can be blended instead of selected by a discontinuous rule [8].

The proposed architecture uses this principle in a modular form. A global ANN provides a direct baseline for the full nonlinear mapping. Three local ANNs specialize in high-performing, normal, and stressed operating zones. A fuzzy fusion layer then weights the local outputs according to the current state. This design preserves the practical interpretability of regime-specific models while avoiding the instability and high variance that can arise when a hard threshold determines which model is active [9], [10]. Recent intelligent warehousing research increasingly combines specialized learning mechanisms with dynamic state information, reinforcing the need to evaluate both predictive accuracy and robustness under changing operating conditions [21].

The distinction from earlier fuzzy-clustering studies [2] is deliberate. Fuzzy clustering there is used to group picker characteristics and reduce the number of ANN inputs before predicting picking time. Here, fuzzy membership is used as an output-fusion mechanism for multiple local predictors, and the predicted object is a vector of interdependent warehouse KPIs. The contribution is therefore architectural and task-oriented rather than a claim that fuzzy logic is universally superior to other machine-learning methods.

C. Simulation and Decision Support in Warehouse Operations

Prediction alone does not determine whether a warehouse policy will perform well. Research on dynamic order picking further shows that warehouse performance is shaped by the relationship between order characteristics, operational decisions, and changing system conditions. Therefore, predictive models are likely to be most useful when integrated into broader decision-support frameworks that compare different operational policies, rather than used as standalone forecasting tools [15]. A forecast becomes operationally useful when it can be evaluated under alternative staffing, batching, routing, release, replenishment, or scheduling decisions. Ngunjiri [11] presents a quantitative, simulation-based framework wich integrate machine learning predictions with supply chain simulation to optimize resilient logistics systems. He describes an integrated machine-learning and simulation framework in which predictive models estimate processing times and bottlenecks, and those predictions are embedded in simulations of alternative warehouse strategies.

The same logic motivates the simulation-based benchmark in this study. The simulator supplies controlled nonlinear dynamics and generates observations across normal and stressed conditions. The neural models then approximate the simulator's future KPIs from lagged operational information. This arrangement makes it possible to test generalization under known regimes and under transitions without conflating model error with uncontrolled variation in a proprietary warehouse dataset.

The simulation approach also has a limitation: performance gains demonstrated on generated data establish approximation capability under the assumed system equations, not immediate industrial validity. Ngunjiri's [11] work emphasizes the value of combining predictive analytics with simulation for complex manual and automated warehouses, but the present study still requires validation against real warehouse-management-system data before its numerical results can be interpreted as evidence of deployment performance.

D. Research Gap and Positioning

The digitization of logistics operations via Internet-of-Things sensors, cloud computing, and real-time enterprise resource planning platforms has transformed supply chain visibility [12]. The reliability of such predictive systems also depends on transparent and accessible logistics information, since incomplete or poorly integrated operational data can limit the effectiveness of supply-chain analytics [20]. The reviewed literature on this field supports three premises: picking time is operationally important, fuzzy information can represent heterogeneity in ANN prediction, and machine learning can be coupled with simulation to evaluate warehouse alternatives [2]. The unresolved problem addressed here is how to combine these ideas into a smooth, modular, multi-output predictor that remains informative when the warehouse moves between operating regimes.

Accordingly, this study does not claim that modular fuzzy ANNs should replace modern deep-learning or optimization methods in every warehouse. It tests a narrower proposition: when a warehouse simulator exhibits distinct nonlinear operating zones, local MLPs combined through fuzzy fusion may offer a favorable balance between predictive accuracy, regime robustness, interpretability, and training cost relative to one global MLP [11]. That proposition is evaluated experimentally in the sections that follow.

E. Positioning Against Modern Deep Learning and Gating Mechanisms

It is necessary to position the proposed fuzzy-modular architecture against modern deep learning paradigms, specifically Mixture of Experts (MoE) and deep gating networks. Mixture-of-Experts architectures combine specialized neural networks through gating mechanisms to model different regions of a complex input space [17]. Modern MoE architectures typically replace fuzzy validity functions with a learned, softmax-based gating network that dynamically routes inputs to specialized sub-networks [9]. While learned gating networks offer high flexibility, they introduce additional parameters that require careful regularization to prevent overfitting, and their decision boundaries are less transparent than explicit fuzzy rules.

By utilizing Gaussian fuzzy membership functions derived from explicit regime centers, the proposed architecture sacrifices some asymptotic flexibility in exchange for strict interpretability, lower data requirements, and guaranteed smooth transitions across regime boundaries. This study does not claim that modular fuzzy ANNs should replace modern deep-learning or learned-gating methods in every warehouse context. Rather, it tests a narrower proposition: when a warehouse simulator exhibits distinct nonlinear operating zones, and data is limited, local MLPs combined through transparent fuzzy fusion may offer a favorable balance between predictive accuracy, regime robustness, interpretability, and training cost relative to one global MLP or a heavily parameterized deep MoE. This proposition is evaluated experimentally in the sections that follow.

III. Warehouse System Model

The modeled system represents a discrete-order fulfillment warehouse in which customer orders arrive over time, are released to picking, transported to a packing/dispatch area, and either complete fulfillment or contribute to backlog. The design follows the operational focus of warehouse prediction studies, where picking time is a key variable for organizing material and information flows. Machine-learning predictions are treated as decision-support inputs to a simulation that can test batching, picker assignment, routing, and task-scheduling policies.

Let the discrete time index be k $\in\{1,2,\ldots,T\}$, with each interval representing a fixed planning period. The warehouse contains Z picking zones, Pk available pickers, and Mk available material-handling resources at time k. The system state vector is defined as:

$$X(k)=[\lambda(k),\ b(k),\ u(k),\ p(k),\ r(k),\ c(k),\ a(k),\ q(k)]^T \quad (1)$$

where, $\lambda(k)$ is the order-arrival rate, $b(k)$ is the open-order backlog, $uP(k)$ is the utilization of picking resources, $p(k)$ is the proportion of priority orders, $r(k)$ is the replenishment-delay rate, $c(k)$ is the congestion level, $a(k)$ is the equipment-availability ratio, and $q(k)$is the number of orders released to picking during interval k.

The simulation produces three primary performance outputs:

$$y(k) = [OF(k),\ PCT(k),\ TH(k)]^T \quad (2)$$

where, $F(k)$ is the order-fulfillment rate, $TP\ (k)$ is the mean picking cycle time, and $\theta(k)$ is warehouse throughput. These outputs capture service completion, processing speed, and productive capacity. A data-driven warehouse-simulation framework similarly treats processing time and bottlenecks as predicted KPIs and evaluates their consequences under alternative operating strategies.

A. Order-Arrival and Backlog Dynamics

Orders arrive according to a time-varying process. For a tractable nonlinear benchmark, the arrival rate is defined as

$$\lambda(k) = \lambda[1 + \alpha s(k) + \alpha ph(k)] + \epsilon\lambda(k) \quad (3)$$

where, $\lambda(k)$ is the nominal arrival rate, $\alpha s(k)$ is a seasonal or promotional intensity, $\alpha ph(k)$ is a demand-shock signal, $\alpha s(k)$ and $\alpha ph(k)$ are sensitivity coefficients, and $\epsilon\lambda(k)$ is a bounded disturbance. The number of arriving orders is

$$A(k) = max\big(0,\ \lambda(k) + \epsilon A(k)\big) \quad (4)$$

The open-order backlog evolves as

$$b(k+1) = max\{0,\ b(k) + A(k) - D(k)\} \quad (5)$$

where, D(k) denotes the number of orders dispatched during interval k. The release quantity is constrained by available labor, equipment, and replenishment readiness:

$$q(k) = min\{b(k) + A(k),\ Pk\,\mu p(k),\ Mk\,\mu m(k),\ Rk\} \quad (6)$$

where, $\mu p(k)$, and $Mk\ \mu m(k)$, are effective picker and material-handling capacities, and Rk is the number of orders whose required inventory is available for release.

B. Picking-Cycle-Time Model

For order i released at time k, the picking cycle time is modeled as:

$$\tau i(k) = \tau 0 + \beta 1ni + \beta 2di + \beta 3c(k) + \beta 4u(k) + \beta 5r(k) - \beta 6ei + \beta 7nic(k) + \epsilon i(k) \quad (7)$$

where, ni is the number of order lines, di is the travel distance, ei is picker experience, and the β coefficients quantify their effects. The interaction term nic (k) introduces nonlinear growth when large orders are processed under congestion. The average picking cycle time is

$$PCT\ (k) = 1Nk\sum i = 1Nk\tau i(k) \quad (8)$$

where, Nk is the number of completed picks in the interval. The effective processing capacity is then

$$C(k) = Pk\,Hk\,\eta p(k)PCT\ (k) \quad (9)$$

where, Hk is the available picking time and $\eta p(k)$ is a productivity factor incorporating fatigue, coordination losses, and task interruptions.

C. Fulfillment and Throughput Equations

Let $OF(k)$ denote the number of orders completed completely and D(k) the number dispatched. The order-fulfillment rate is defined as

$$OF(k) = F(k)max\{1, A(k)\}. \quad (10)$$

A complete order requires successful picking, inventory availability, and packing capacity. Thus,

$$F(k) = min\{C(k),\ Rk,\ Kk\}\,\phi(k), \quad (11)$$

where, Kk is packing/dispatch capacity and $\phi(k)$ is the order-completeness factor. To represent nonlinear interactions among warehouse constraints, define

$$\phi(k)=\sigma((\theta 0-\theta 1r(k)-\theta 2c(k)-\theta 3b(k)+\theta 4a(k)+\theta 5a(k)c(k)), \quad (12)$$

$$\text{with } \sigma\ (v) = \frac{1}{1+E^{-V}} \quad (13)$$

Throughput is

$$TH(k) = D(k)\Delta t \quad (14)$$

where, Δt is the length of the simulation interval.

D. State Transition and Prediction Target

The complete simulator can be represented by the nonlinear state-space system:

$$x(k+1) = f(x(k), u(k), \epsilon(k)), y(k+1) = g(x(k), u(k), \epsilon(k)) \quad (15)$$

where, $u(k)$ contains controllable or exogenous variables such as staffing, release policy, batching size, routing policy, replenishment priority, and equipment availability. The neural predictor receives a regression vector containing current and lagged observations:

$$\xi(k) = [u(k), u(k-1), y(k), y(k-1)]T \quad (16)$$

The prediction problem is to learn an approximation from simulated input-output observations. All continuous variables are normalized to [0,1] before training and transformed back to their physical units for reporting.

E. Composite Order-Fulfillment Performance Indicator

To complement the three-output prediction task, a composite warehouse service indicator is defined as:

$$WOFI(k) = w1OF(k) + w2TH\sim(k) + w3[1 - PCT\sim(k)] \quad (17)$$

subject to

$$w1 + w2 + w3 = 1, wj \geq 0. \quad (18)$$

The tilde denotes min-max normalization using the training range. Unless operational priorities dictate otherwise, the initial experiments can use $w1$=0.5, $w2$=0.3, and $w3$=0.2, giving greatest weight to complete order fulfillment. WOFI is used for regime definition and summary evaluation rather than replacing the underlying KPIs.

F. Global Neural Model

The baseline is a direct, single MLP trained on all simulated operating regimes. For an input vector, the hidden-layer activation is

$$h(k) = \sigma(W_H\,\xi(k) + b_H) \quad (19)$$

and the output prediction is

$$\hat{Y}_G(k+1) = W_O(k) + bo \quad (20)$$

A linear output layer is used because the normalized outputs are continuous. The global model minimizes the mean squared prediction error

$$\text{JG} = \frac{1}{N}\sum_{K=1}^{N} \| \text{y(k+1)} - \hat{Y}_G\ (K+1) \|_2^2 \quad (21)$$

The model may be trained with backpropagation or Levenberg–Marquardt, using the same optimization procedure for all competing architectures so that the comparison isolates the effect of modularization and fuzzy fusion.

The simulated observations are partitioned into three operating regimes using WOFI and workload conditions:

$$Z1: WOFI \geq \gamma_H$$
$$Z2: \gamma_L < WOFI < \gamma_H \quad (22)$$

$$Z3: WOFI \leq \gamma_L$$

Here, $Z1$ represents high-performing, low-congestion operation; $Z2$ represents normal or moderately loaded operation; and $Z3$ represents stressed operation with backlog, congestion, replenishment delays, or equipment loss. To avoid discontinuities at the boundaries, the hard partition is used only to construct local training sets; fuzzy membership is used during prediction.

Each local MLP has the same input and output definitions as the global model but is trained on its corresponding:

$$\widehat{y_j}(k+1) = W_{O,j}\sigma(W_{h,j}\xi(k) + b_{h,j}) + bo, j, \text{j}\in\{1,2,3\}. \quad (23)$$

The local objective is

$$Jj = \frac{1}{N_J}\sum_{K\in Z_j} \| y(k+1) - \widehat{y_j}(k+1) \|_2^2 \quad (24)$$

G. Fuzzy Fusion Mechanism

For each regime, define $\hat{v}y(k)$ as a center from the training observations in that regime. The unnormalized Gaussian validity of local model j is:

$$\widehat{v_j}(k) = exp[-\frac{1}{2}\big(\varepsilon(k) - \mu_j\big)^T \sum_j^{-1}(\varepsilon(k) - \mu_j)] \quad (25)$$

where, controls the width and orientation of the membership region. The normalized validity weights are:

$$v_j(\text{k}) = \frac{\widehat{v_j}(k).}{\sum_{l=1}^{3} \widehat{v_l(k)}} \quad (26)$$

This preserves continuity when the warehouse moves between normal and stressed conditions and allows the model to exploit regime-specific approximations without imposing a hard switch. The use of fuzzy information to support ANN prediction is consistent with prior warehouse work that uses fuzzy clustering to represent heterogeneous picker characteristics before neural prediction.

IV. Simulation and Evaluation Results

The experiment should generate separate training, validation, and test trajectories. Test trajectories must include both familiar regimes and transitions between regimes, because a model that performs well only within isolated zones may still fail during congestion onset or recovery. The principal metrics are:

$$RMSE = \sqrt{\frac{1}{N}\sum_{K=1}^{N} \| y(k) - \hat{y}(k) \|_2^2} \quad (27)$$

$$MAE = \frac{1}{N}\sum_{K=1}^{N} \| y(k) - \hat{y}(k) \|_1 \quad (28)$$

and the composite WOFI error. Training time, number of hidden neurons, parameter count, and convergence iterations should be recorded for the global model and each local model. The total modular training time is

$$TF = \sum_{J=1}^{3} T_j \quad (29)$$

while inference time includes the three local forward passes, membership calculations, and weighted fusion. This distinction is necessary: shorter local training does not automatically imply lower real-time inference cost.

The planned results table should therefore report global performance, each local model's within-regime performance, fused performance over the complete test set, transition-regime performance, and computational cost. No numerical performance claim should be made until the simulator has been implemented and the experiments executed.

A. Experimental Implementation and Evaluation Results

A reproducible nonlinear simulator was executed with four 900-step training trajectories and two 900-step held-out trajectories. After one-step alignment, the benchmark contained 3,596 training examples and 1,798 test examples. Each observation contained ten predictors: current arrival, backlog, congestion, replenishment delay, equipment availability, release quantity, and four one-step lagged variables. The targets were order-fulfillment rate, picking cycle time, and throughput. A fixed random seed (42) was used for data generation and model initialization.

The global baseline and each of the three local models used a single eight-neuron logistic hidden layer and standardized inputs and outputs. Local training sets were defined by the 33rd and 67th training WOFI percentiles. During inference, the local predictions were blended with Gaussian weights computed from standardized input-space distances to the three regime centers. No test observations were used to define the regime thresholds, centers, or preprocessing parameters.

B. Result

On the held-out trajectories, the global MLP achieved standardized RMSE values of 0.670 for the fulfillment rate, 1.013 for picking cycle time, and 0.391 for throughput (Fig.1).

TABLE I. Predictive Performance Comparison on Held-Out Trajectories

Architecture	Fulfillment Rate RMSE	Picking Cycle Time RMSE	Throughput RMSE
Global MLP	0.670	1.013	0.391
Fuzzy-Modular	1.171	1.208	0.732
Degradation (%)	+74.8%	+19.2%	+87.2%

The fuzzy modular model achieved 1.171, 1.208, and 0.732, respectively (Table 1). These per-KPI error metrics are summarized in Fig. 1. Thus, in this reproducible run, fuzzy fusion did not improve aggregate predictive accuracy: mean RMSE increased by 49.9% and mean MAE increased by 29.9% relative to the global baseline.

The degradation is concentrated in the regime-specific architecture rather than in a single output. The local models were trained on fewer observations, and their Gaussian centers were estimated from a heterogeneous input space; consequently, the fused estimate can be pulled toward a poorly calibrated local prediction even when the global model remains stable. This serves as a useful negative control, demonstrating that modularization should not be treated as inherently beneficial without transition-aware validation and careful tuning of regime definitions, membership widths, and local sample sizes.

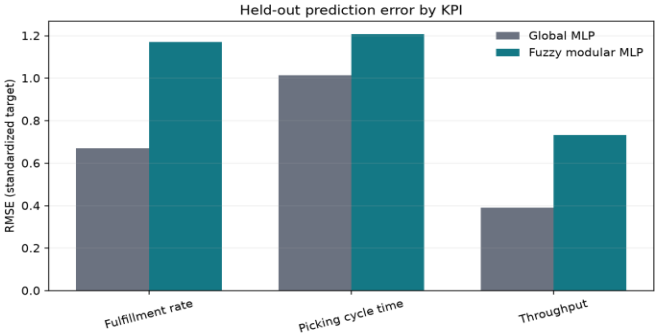


Fig. 1. Held-out standardized RMSE by KPI.

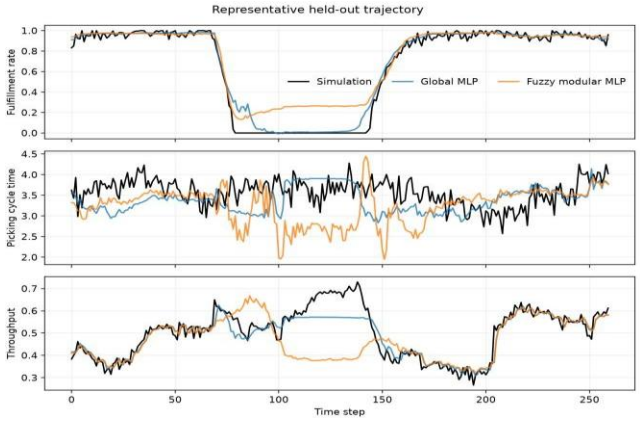


Fig. 2. Representative held-out trajectory.

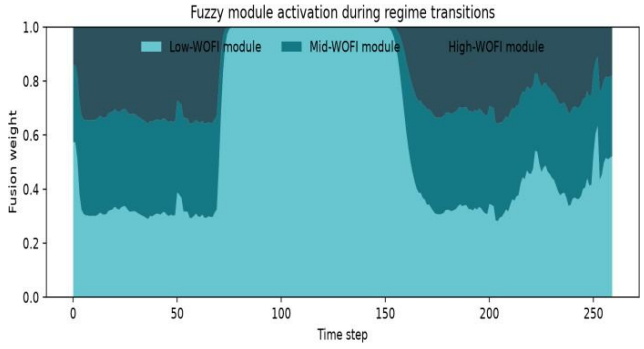


Fig. 3. Gaussian fusion weights across a held-out trajectory.

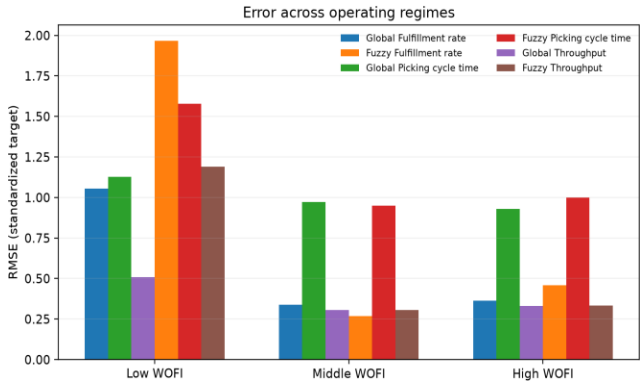


Fig. 4. Prediction error across low-, middle-, and high-WOFI test regimes.

C. Evaluation Interpretation

The representative held-out trajectory (Fig. 2) shows that both predictors follow the broad simulated dynamics, but the fuzzy model exhibits larger deviations during and after workload shocks. Concurrently, the Gaussian fusion weights across the trajectory (Fig. 3) confirm that the model's activation changes smoothly rather than switching discontinuously. However, as the aggregate error metrics demonstrate, mathematical smoothness alone does not guarantee predictive accuracy. Finally, the prediction error stratified across low-, middle-, and high-WOFI test regimes (Fig. 4) provides a detailed view of where the modular model struggles, indicating whether future improvements should focus on better local specialization or a more reliable global baseline.

The current experiment supports a reproducible benchmark and establishes a clear evaluation protocol, rather than making a direct deployment claim. Before drawing operational conclusions, future studies should repeat the experiment over multiple random seeds, report confidence intervals across trajectories, tune membership widths using validation data only, and compare against simpler regime-aware baselines (such as a hard-switch model) and a mixture-of-experts model with learned gating. Furthermore, real warehouse-management-system (WMS) data are required to assess external validity.

D. Limitations and Future Work

Although this study provides a rigorous and reproducible benchmark for predicting multiple warehouse-performance measures, it has several limitations. First, the evaluation was conducted entirely in a simulated environment. While the simulator represents important warehouse dynamics, such as congestion, replenishment delays, and capacity constraints, results generated from assumed system equations cannot, by themselves, demonstrate that the models will perform equally well in an industrial setting. Second, the experiments used a single random seed for both data generation and model initialization. This limits the extent to which the reported error metrics can be generalized. Finally, the weaker performance of the fuzzy-modular architecture points to a specific methodological issue. In a high-dimensional and heterogeneous input space, using simple distance-based Gaussian fusion can give too much influence to poorly calibrated local models, particularly when only a small number of training examples are available for a given operating regime.

Future research will address these limitations in three main ways. First, the experiments will be repeated using multiple random seeds, with mean performance and 95% confidence intervals reported across a broader range of operational scenarios. Second, the architecture will be improved by replacing fixed Gaussian validity functions with learned, data-driven gating mechanisms, such as those used in mixture-of-experts models. Membership-function widths will also be tuned using a dedicated validation set to reduce the risk of overfitting regime boundaries. Third, and most importantly, the framework will be tested using high-frequency data from real Warehouse Management Systems.

This step is essential for determining whether regime-aware modularization can improve predictive accuracy, robustness during operational transitions, and decision support in complex warehouse environments beyond the simulated setting.

V. CONCLUSION

This study presented a simulation-based, multi-output predictive framework for warehouse performance, benchmarking a global multilayer perceptron (MLP) against a fuzzy-integrated modular neural architecture. Experimental evaluation revealed that the modular model underperformed the global baseline, with aggregate RMSE increasing by 49.9%. Rather than invalidating modular prediction, this outcome highlights the critical challenges of applying naive Gaussian fusion in high-dimensional, heterogeneous state spaces. It demonstrates that mathematical smoothness across regime boundaries does not guarantee predictive accuracy without rigorous local calibration and sufficient training data.

Despite this comparative performance outcome, the study establishes three foundational contributions: 1) a holistic multi-output prediction task capturing interdependent warehouse dynamics; 2) a fully specified, reproducible nonlinear simulator; and 3) a rigorous evaluation protocol that explicitly isolates within-regime accuracy from transition-phase robustness. Future work will address the identified limitations by replacing distance-based Gaussian validity functions with learned gating mechanisms and validating the framework against real-world warehouse management system (WMS) data to assess industrial deployment readiness.

FUNDING

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. 265120].

REFERENCES

[1] J. J. Bartholdi III and J. P. Hackman, Warehouse & Distribution Science, First draft: May 22, 1998; revised August 20, 2006.

[2] Kabak, Ö., Güleç, N. Data-Driven Approach to Order Picking Time Prediction Using Fuzzy Clustering and ANN. In: Kahraman, C., Cebi, S., Cevik Onar, S., Oztaysi, B., Tolga, A.C., Sari, I.U. (eds) Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation. *INFUS. Lecture Notes in Networks and Systems*, vol 307. Springer, Cham. 2021. Doi: 10.1007/978-3-030-85626-7_3

[3] Frikha M.A. and Mraihi R. "Audit of Authorised Economic Operators (AEO): A new risk management approach based on Logistics Inoformation System (LIS) analysis*", 2020 IEEE, 13 th international Conference on Logistics* (LOGISTIQUA), 2020. 10.1109/LOGISTIQUA49782.2020.9353944

[4] Wenrong Lu, Duncan McFarlane, Vaggelis Giannikas, Quan Zhang, An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research*, vol 248, no. 1, pp. 107-122, ISSN 0377-2217, 2016. Doi: 10.1016/j.ejor.2015.06.074.

[5] J. A. Riedel, R. Facilities planning – 4th edition by J.A. Tompkins, J.A. White, Y.A. Bozer and J.M.A. *Tanchoco. International Journal of Production Research*, vol. 49, no. 24, pp. 7519–7520. Doi : 10.1080/00207543.2011.563164.

[6] Rodrigo Furlan de Assis, Alexandre Frias Faria, Vincent Thomasset-Laperrière, Luis Antonio Santa-Eulalia, Mustapha Ouhimmou, William de Paula Ferreira. *Machine Learning in Warehouse Management: A Survey, Procedia Computer Science*, Vol 23, pp 2790-2799, ISSN 1877-0509, 2024. Doi: 10.1016/j.procs.2024.02.096.

[7] Md Redwanul Islam, Md Nahid Hossain, & Md. Zahid Hasan Tusar. predictive analytics in supply chain management a review of business analyst-led optimization tools. *Review of Applied Science and Technology*, vol. 6, no. 1, pp. 34-73. 2021. Doi: 10.63125/5aypx555

[8] J. -S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May-June 1993. Doi: 10.1109/21.256541.

[9] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, Geoffrey E. Hinton; Adaptive Mixtures of Local Experts. *Neural Comput*; vol. 3, no 1, pp 79-87, 1991. Doi: 10.1162/neco.1991.3.1.79

[10] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean. Outrageously Large Neural Networks. *The Sparsely-Gated Mixture-of-Experts Layer*, vol. 1, 2017. DOI: 10.48550/arXiv.1701.06538.

[11] Justus Ngunjiri, artificial intelligence-driven supply chain optimisation: integrating predictive analytics for resilient logistics systems. *Gulf Journal of Advance Business Research*. Vol. 1 No. 3, 2023.

[12] Frikha, M. A., Mrad, M., Boujelbene, Y., & Soliman, Y. M. A. Hybrid State Estimation and Recurrent Neural Networks for Supply Chain Disruption Detection. *International Journal of Advanced Computer Science and Applications*, vol 17, no 7, 2026. Doi: 10.14569/IJACSA.2026.0170789.

[13] De Assis, R. F., Faria, A. F., Thomasset-Laperrière, V., Santa-Eulalia, L. A., Ouhimmou, M., & de Paula Ferreira, W.. Machine learning in warehouse management: A survey. *Procedia Computer Science, vol 232*, pp 2790–2799, 2024. Doi: 10.1016/j.procs.2024.02.096

[14] Kabak, Ö., & Güleç, N.. Data driven approach to order picking time prediction using fuzzy clustering and ANN. In C. Kahraman, S. Cebi, S. C. Onar, B. Oztaysi, A. C. Tolga, & I. U. Sari (Eds.), *Intelligent and fuzzy techniques for emerging conditions and digital transformation* (Lecture Notes in Networks and Systems, Springer, vol. 307. 2022. Doi: 10.1007/978-3-030-85626-7_3

[15] Lu, W., McFarlane, D., Giannikas, V., & Zhang, Q. An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research*, vol. 248 no. 1, pp 107–122, 2016. Doi: 10.1016/j.ejor.2015.06.074

[16] J. -S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May-June 1993. Doi: 10.1109/21.256541.

[17] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E.. Adaptive mixtures of local experts. *Neural Computation,vol. 3, no.* 1, pp. 79–87, 1991. Doi: 10.1162/neco.1991.3.1.79.

[18] M. A. Frikha, M. Mrad, Y. Boujelbene, and S. Ben Othman, "Deep hybrid learning for sustainable industrial forecasting: Integrating CNN–LSTM models to enhance economic efficiency and carbon performance, *International Journal of Advanced Computer Science and Applications*, vol. 17, no. 2, 2026. Doi: 10.14569/IJACSA.2026.0170264.

[19] Rad, F.F., Oghazi, P., Onur, İ. *et al.* Adoption of AI-based order picking in warehouse: benefits, challenges, and critical success factors. *Rev Manag Sci* 19, pp. 3495–3540, 2025. Doi: 10.1007/s11846-025-00858-1.

[20] M. A. Frikha, "Sustainable and transparent global supply chains through blockchain-enabled logistics information systems," *Quality-Access to Success*, vol. 27, no. 213, pp. 46–54, 2026. Doi: 10.47750/QAS/27.213.05.

[21] X. Wang, L. Zhang, L. Wang, E. Ruiz Zuñiga, X. V. Wang, and E. Flores-García, Dynamic multi-tour order picking in an automotive-partwarehouse based on attention-aware deep reinforcement learning, *Robotics and Computer-Integrated Manufacturing*, vol. 94, p. 102959, 2025. Doi: 10.1016/j.rcim.2025.102959.

Benchmarking ML-Based DDoS Detection Algorithms: Efficiency, Effectiveness, and Generalization

Zhilbert Tafa*, Aurora Uka
University for Business and Technology, Prishtina, Kosovo

***Abstract*—Distributed Denial of Service (DDoS) attacks are among the most significant cybersecurity threats today. Traditional DDoS detection systems struggle to cope with emerging attack vectors, while Deep Learning (DL)-based approaches are limited by their high computational complexity. Shallow Machine Learning (ML) techniques have been widely explored, either as standalone solutions or to complement DL approaches. However, the existing literature remains ambiguous regarding the trade-offs among classification accuracy, computational complexity, generalization, and the consistency of ML-based methods across different network environments. In this work, we compared six representative models spanning supervised, semi-supervised, and unsupervised ML. We used five datasets to assess their DDoS identification capability, computational efficiency, generalization, and cross-domain performance. XGBoost demonstrated the best balance between DDoS identification capability (F1 = 0.984) and computational efficiency. Isolation Forest (IF) showed solid discriminative performance (F1 = 0.673) at a very low computational cost, making it suitable for resource-constrained devices. Other examined methods excel in certain aspects while remaining limited in other metrics. All models exhibited strong generalization performance. Apart from Stochastic Gradient Descent One Class Support Vector Machine (SGD-OCSVM), all other models showed high consistency in DDoS identification across different network setups.**

***Keywords*—Anomaly detection; autoencoder; DDoS; isolation forest; LOF; machine learning; OCSVM; RF; XGBoost**

I. Introduction

Distributed Denial-of-Service (DDoS) attacks are among the most prevalent cyberattacks in recent years, causing severe service disruptions or degradation. Their potentially harmful repercussions for organizations, governments, and individuals emphasize the importance of robust detection and mitigation strategies [1].

DDoS attacks have reached an unprecedented level, with vast growth in volume, intensity, frequency, and attack vectors [2]. In a DDoS attack, the attackers typically use botnets of compromised networked devices (e.g., computers, servers, IoT devices, etc.) to coordinate attacks through a command-and-control (C2) server (Fig. 1).

DDoS threats are commonly categorized into two main groups: volumetric and low-rate (stealthy) attacks.

*Corresponding author.



Fig. 1. DDoS attacks orchestrated using C2 servers and botnets.

Volumetric attacks are usually launched by flooding the network infrastructure with an intense number of packets, which can result in the consumption of network resources [3]. On the other hand, low-rate attacks typically target the application layer. Through long-lasting sessions, they mimic legitimate user behavior, which makes them particularly hard to detect [4].

With the emergence of new technologies, computer networks are becoming more structurally complex, diverse, and heterogeneous [5]. This expands the attack surface, increasing the likelihood of sophisticated, multi-vector attacks that combine volumetric and application-layer exploits [6]. Moreover, the rapid advancements in the field of artificial intelligence (AI) have made attacks easier to launch and harder to detect. Current models for analyzing DDoS attacks are often inadequate for addressing the complexity of emerging DDoS campaigns [7].

To overcome the limitations of traditional defense strategies, modern Intrusion Detection and Prevention Systems (IDPSs) have increasingly incorporated intelligent adaptive decision-making mechanisms. However, as these systems are often devised to operate across resource-constrained devices, achieving a balance between low computational complexity and high classification performance remains a challenge. While Deep Learning (DL) methods excel in classification performance, they require significant computational resources and a large amount of training data. Hybrid IDPS may include only lightweight models, or be combined with DL methods, depending on the application.

With the exception of the autoencoder (for anomaly detection), in this study, only representative shallow ML algorithms that may be useful for incorporation into the hybrid system are considered. We developed a benchmarking framework to evaluate prominent supervised, unsupervised, and semi-supervised algorithms for DDoS detection in terms of

classification performance, computational efficiency, generalization, and performance consistency across different network environments.

To the best of our knowledge, this study is the first to provide a broad comparative analysis, offering insights into the most suitable ML methods for network traffic classification.

II. Related Work

Academic research has shown a high level of interest in identifying combinations of data-driven models that best suit DDoS detection in diverse implementation scenarios, exploring statistical, entropy-based, ML, and DL methods. As our work focuses on "lighter" data-driven models, this section reviews recent and relevant studies in this domain.

Trigui et al. [8] measure training time, inference latency, peak memory usage, and model size of Extreme Gradient Boosting (XGBoost), Random Forest (RF), AdaBoost, K-Nearest Neighbours (KNN), Naïve Bayes (NB), and Support Vector Machine (SVM) models across two datasets. Although the models achieve impressive results, the choice of evaluation datasets may not reflect the real-world traffic environment.

In [2] authors evaluated RF, XGBoost, NB, Logistic Regression, Linear SVM, Decision Tree (DT), and Hidden Markov Models (HMM) against the LR-HR DDoS 2024 and the INSDN datasets. The optimized XGBoost model achieved the best performance. However, the analysis was limited to SDN environments and relied on a restricted set of features.

Hirsi et al. [9] use RF as a feature selection model and SVM for classification of SDN traffic. The models were evaluated on a self-generated SDN-DDoS dataset and NSL-KDD, outperforming standalone SVM, NB, FNN, and RF models. However, NSL-KDD is not representative of SDN traffic and is now considered outdated, which limits the applicability of the results.

An unsupervised approach combining blockchain technology, Isolation Forest, and three feature-selection methods was proposed in [10]. It outperforms models that use individual feature-selection techniques. However, evaluation solely on the (outdated) CICDDoS2019 limits its generalizability.

Das et al. [11] evaluated five supervised and five unsupervised algorithms across three datasets. They proposed a hybrid ensemble model combining supervised and unsupervised methods, achieving promising results in both standalone and hybrid configurations. The study is limited by outdated data and an inconsistent feature set across the compared models.

Reference [12] employs K-Means, GMM, and Stochastic Gradient Descent One Class SVM (SGD OCSVM) on the CICDDoS2019 dataset for zero-day attack detection, and Gaussian Random Projection (GRP) for feature selection. The proposed combination demonstrates high accuracy, but it is limited to only a specific and somewhat outdated networking environment.

Hybrid models often combine supervised methods for their high accuracy with unsupervised methods for their ability to detect novel attacks and adapt to unseen patterns. Recent approaches have used DL models combined with unsupervised models to effectively detect zero-day attacks [13].

It is evident that the literature lacks a thorough analysis of less complex (shallow ML) models across diverse networking conditions, comparing them under the same or very similar assumptions. This work aims to address this gap by benchmarking six representative supervised and anomaly detection approaches. The evaluation was performed across datasets acquired from traditional computer networks, IoT, SDN, and Cloud infrastructure.

III. Materials and Methods

Fig. 2 shows the flowchart of the proposed methodology. The acquired datasets were first cleaned and encoded. Hyperparameters for all models were optimized using inner 3-fold cross-validation, and the best settings, selected based on average F1 score, were then applied for training with outer 5-fold cross-validation. Feature selection/extraction and data scaling were performed within each training cross-validation fold to prevent data leakage.

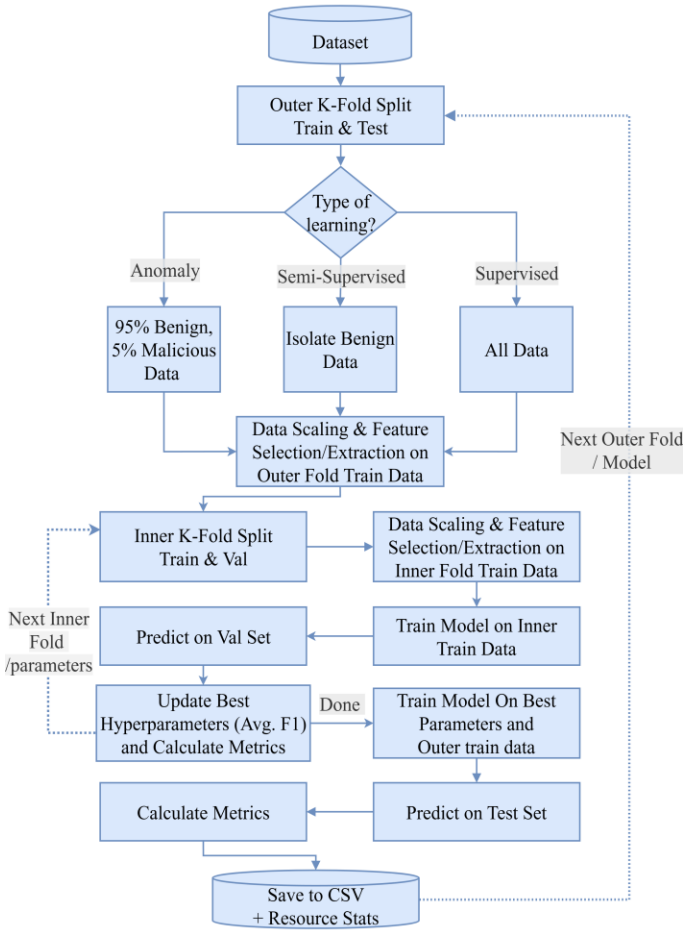


Fig. 2. ML pipeline for DDoS detection.

Preprocessing for different models involved different approaches. The data for supervised models were standardized using the Min-Max statistical method, whereas unsupervised/anomaly models used the Z-Score standardization method.

The dimensionality of the datasets was reduced using Chi-Squared feature selection for supervised models and Principal Component Analysis (PCA) feature extraction for the other machine learning models, excluding the autoencoder model, for which no feature selection or extraction method was used. This approach of combining MinMax with Chi-Squared is essential, as Chi-Squared cannot be computed on negative data, whereas Z-Score can produce negative values but preserves the variance of the features. PCA, on the other hand, is sensitive to scaling and does not require the target feature, making it suitable for unsupervised models. A fixed random seed of 42 was used for sampling data during outer and inner folds, as well as for PCA feature extraction.

To execute the ML pipeline, we used a Hetzner Cloud dedicated server with 4 vCPUs, 16 GB RAM, and 160 GB storage. The 4 vCPUs were based on an AMD EPYC Milan processor with a 64-bit x86_64 architecture. The server ran the Ubuntu 24.04.4 LTS (Linux kernel 6.8.0-101-generic) operating system. The programming environment and the respective list of libraries are presented in Appendix A (Table IV).

The following subsections provide a detailed description of each step of the DDoS detection model development.

A. Datasets

The datasets were selected based on several criteria. Our goal was to cover diverse network traffic patterns, DDoS attack types, and networking frameworks. Therefore, the datasets include traffic from conventional networks, IoT environments, SDN controllers, and Cloud environments. We also balanced recency with adoption in the literature. However, in this process, some widely used datasets were excluded due to insufficient documentation on the data acquisition setup or outdated content. We finally limited the scope of the materials to the five datasets summarized in Table I.

TABLE I. DATASETS

Dataset	Year	Environment	Records	Features	Key Strength	Main Limitation
ASEADOS SDN IOT [14]	2026	SDN + IoT	457,044	84	Slow-rate DDoS attacks	DDoS attacks mainly with stress test tools
BCCC-cPacket-CloudDDoS-2024 [15]	2024	Cloud	540,494	319	Large feature set	DDoS methods unspecified. Data imbalance
SDN Dataset [16]	2020	IoT + SDN	27.9M / 30.2M	34	Large-scale dataset	Duplicates; undocumented DDoS attack types
TON-IoT [17]	2020	IoT + Cloud + SDN	830,302	47	Traffic diversity	DDoS attacks are unspecified, despite using UFOnet and custom scripts.
INSDN [18]	2020	SDN	343,889	84	SDN-specific and slow rate attacks	Duplication after removing the irrelevant features

B. Data Preprocessing

A data cleaning audit was performed on all datasets. The sensitivity analysis of the deduplication strategy showed a non-significant reduction in precision and recall after data deduplication, except for the dataset obtained from the cloud environment, where a small increase in both metrics was observed. This implies that deduplication is safe in all cases.

Label encoding was applied to the label and protocol columns, assigning a unique integer to each class. To prevent ordinal bias, source and destination ports were transformed into frequency-based and port-group features (Unknown, Well-known, Registered, and Private). The resulting port-group features were then encoded using One-Hot encoding.

To prevent data leakage, standardization and feature selection were performed within each fold of the k-fold cross-validation process, while the validation and test data remained unseen during these steps (Fig. 2). For supervised models, Min–Max normalization was applied, while feature selection was performed using the Chi-square (χ^2) method. For unsupervised methods, Z-Score standardization was performed before PCA feature extraction because PCA is sensitive to the scale and variance of the input features. Feature selection and extraction reduce dataset dimensionality and improve both model efficiency and classification performance.

Min-Max is a data standardization statistical method used for converting data within a range [0,1]. Min-Max normalization is calculated as follows:

$$X_{scaled} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

where:

X is an original value from the dataset.

X_{min}, X_{max} represent the minimum and maximum values of a given feature, respectively.

Z-score is also a statistical data standardization method used to convert data so that the features contribute equally to the analysis. Z-Score can be described by Eq. (2):

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

where:

$x -$ observed value

$\mu -$ mean

$\sigma -$ standard deviation

These methods are necessary for model training because features with larger numerical ranges can influence the model and create a biased prediction.

Chi-squared feature selection is a statistical method used to select features that are correlated with the target feature. The Chi-Squared statistic is calculated via the χ^2 value:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (3)$$

where:

O_i – observed frequency

E_i – expected frequency

χ^2 – Chi-Squared statistic

Higher values of χ^2 indicate greater relevance.

Since the Chi-square (χ^2) method requires non-negative feature values, Min–Max normalization is applied beforehand.

The elbow method was used to determine the optimal number of features by identifying the point beyond which additional features provide negligible improvement. The process of identifying the optimal number of features by using the elbow methodology is presented as pseudocode in Algorithm 1.

Principal Component Analysis (PCA) is a statistical method used for feature extraction. It reduces the number of features transforming the feature space into new uncorrelated linear components. This reduces the complexity of the dataset and improves training efficiency.

Algorithm 1: Elbow Method for selecting optimal number of features for χ^2 and PCA

```
Input:
    X_train: Training features,
    y_train: labels,
    model: model type,
Output:
    k: Optimal number of features/components,
If the model is supervised, then
     Compute χ² score for each feature
          using (X_train, y_train)
     Rank features according to their χ² scores
     Compute cumulative feature importance
     Generate cumulative importance curve
      x ← Feature indices,
      y ← Cumulative feature importance
else
     Fit PCA on X_train
     Compute explained variance ratio
     Compute cumulative explained variance
     Generate cumulative variance curve
     x ← Component indices,
     y ← Cumulative explained variance
end if
Determine the knee point using KneeLocator with:
     curve ← "concave"
     direction ← "increasing"
      k ← KneeLocator(x, y, curve, direction)
return k
```

Fig. 3 and Fig. 4 show examples of elbow point identification within one fold of the outer 5-fold cross-validation process for χ^2 feature selection and PCA feature extraction, respectively.

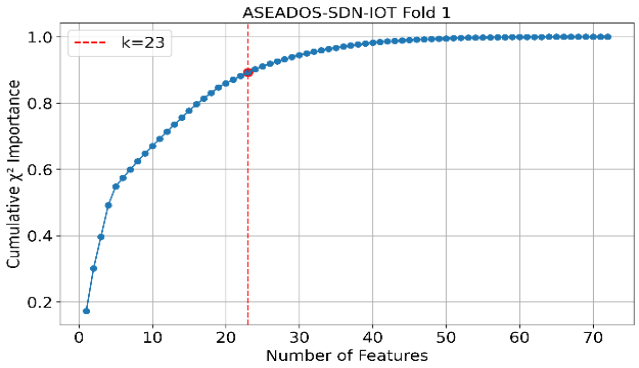


Fig. 3. An Example of the selected number of features for χ^2 feature selection (ASEADOS-SDN-IOT at fold 1).

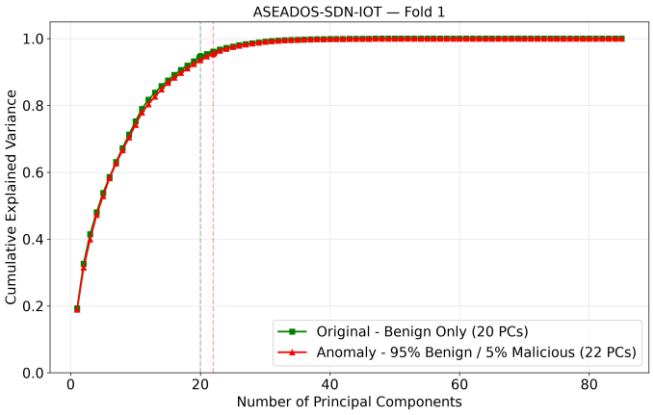


Fig. 4. Example of the selected number of features for PCA feature extraction (ASEADOS-SDN-IOT at fold 1).

C. Model Selection and Implementation

Although DL has become dominant in data science, practical deployments of DDoS detection mechanisms often require a balance between effectiveness and efficiency. Consequently, combined lightweight ML approaches, as well as hybrid approaches that combine DL with traditional ML techniques for DDoS identification, have been extensively explored in the literature. To evaluate the potential of traditional ML techniques in such settings, we selected prominent supervised, unsupervised, and semi-supervised methods that are widely used in DDoS detection. In this subsection, we briefly present their essence.

Supervised ML algorithms operate on labeled data. In our analysis, we included Random Forest (RF) and Extreme Gradient Boosting (XGBoost). These algorithms were selected based on their strong performance in DDoS detection reported in the surveyed literature, outperforming other methods across a wide range of evaluation metrics.

RF randomly divides the feature set into a specific number of decision trees and trains them on different subsets of data. Then, the output is classified through majority voting. The algorithm is robust, resistant to overfitting, and performs well on unbalanced data.

XGBoost uses bootstrapping as an ensemble method, i.e., each decision tree is trained on the residuals of the one before it. This iterative process improves accuracy and enables effective

modeling in a complex feature space. XGBoost is robust to missing values and can handle imbalanced datasets effectively.

Unsupervised ML operates on unlabeled data. In the context of DDoS detection, statistical and unsupervised methods are essential for detecting unseen (zero-day) attacks. Semi-supervised approaches learn on partially labeled benign data, and then detect patterns that deviate from benign behavior.

Based on their results from comparative analyses, we have chosen to examine Local Outlier Factor (LOF) and Isolation Forest (IF) for unsupervised anomaly detection, and autoencoder and SGD-OCSVM for semi-supervised training. Although autoencoders belong to the DL group of algorithms, we have included them in our study because autoencoders are often used for anomaly detection and are therefore of interest for comparison with the other presented anomaly detection algorithms.

LOF is a density-based method that measures the local isolation of data points compared to their neighbors. Specifically, if a data point is surrounded by a lower density compared to the average density of its neighbors, that point is considered an anomaly. This approach is particularly effective in identifying local anomalies.

IF employs a tree-based structure similar to RF. Each tree recursively partitions the data using randomly selected features and split thresholds. The average path length of data points across the trees is evaluated. Since anomalous instances are typically easier to isolate, they tend to have shorter average path lengths than legitimate traffic and are therefore identified as outliers.

An autoencoder is a DL structure typically used for feature extraction and anomaly detection. It consists of an encoder and a decoder. In the context of DDoS detection, the encoder encodes the pattern learned from benign data. Then, the decoder tries to reconstruct the original features from the encoded ones. If the data are malicious, the reconstruction error will be high.

OCSVM is an anomaly detection algorithm based on SVM, except that it is trained exclusively on unlabeled data to find a hyperplane that encloses the majority of benign data. When applied to mixed traffic, anomalies are differentiated as belonging to the other side of the hyperplane. In this work, the surveyed literature motivated us to use SGD OCSVM.

We trained supervised learning algorithms on the mixed benign and malicious data, while semi-supervised models were trained only on benign samples. The autoencoder used a fully connected encoder-decoder architecture with two ReLU-activated hidden layers and a linear output layer. The model was trained using Adam optimization and MSE loss with early stopping, where patience was 5 to prevent overfitting. The autoencoder model creation algorithm is provided in Appendix B (Table V). The anomaly detection models were trained on a subset comprising 95% benign data and 5% malicious data, randomly selected from each training fold. All algorithms were tested on mixed benign and malicious samples. For LOF, IF, and OCSVM, the labels were obtained from their prediction methods, where outliers (-1) were mapped to the malicious class, and inliers (1) were mapped to the benign class.

Autoencoder, on the other hand, identified anomalies using a reconstruction-error threshold tuned to the 95th, 97th, or 99th percentile. Parameters used for model implementation and hyperparameter tuning, along with their selected values for all models, are presented in Appendix C (Table VI).

D. Evaluation Objectives

Our evaluation objectives focused on measuring the models' effectiveness, computational efficiency, generalization, and performance consistency across different network environments. We also analyzed the datasets' content and their discriminatory capacity.

Effectiveness was measured through standard classification metrics, such as accuracy, precision, recall, and F1.

Accuracy measures the proportion of the correctly classified instances out of the total predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where:

TP – True Positive is the sum of all instances correctly classified as belonging to the positive class.

TN – True Negative is the sum of all instances defined as benign traffic by the model and are actually benign traffic.

FP – False Positive is the sum of all instances defined as DDoS attack traffic by the model, but are actually benign traffic.

FN – False Negative is the sum of all instances defined as benign traffic by the model, but are actually DDoS attack traffic.

Precision measures the ratio of the true positives to the total positives. In our case, it represents how many of the predicted attacks were actual attacks.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

Recall, sometimes referred to as sensitivity, measures the proportion of all the correctly predicted DDoS attacks among all the DDoS attacks in the whole class of DDoS attacks.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

F1-score is the harmonic mean between precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Computational efficiency was measured through training time, tuning time, CPU utilization, and peak memory requirements. CPU utilization was calculated as the average value across four CPU cores. These measurements were performed on the same platform to allow for comparison under the same assumptions. However, computational efficiency should not be interpreted in terms of absolute values, as it is platform-specific. Therefore, the results serve only for comparative analysis.

Generalization was measured by the ability of the models to classify unseen data in different folds of the cross-validation process.

Performance consistency across different network environments was measured using the standard deviation of models' classification performance across the datasets.

Discriminatory capacity within datasets was measured through mean normalized confusion matrices, which are suitable for evaluating model performance across multiple groups. In this case, we measured the performance of different models applied to each dataset and calculated mean normalized confusion matrices with relative (percentage) values.

IV. Results

We divide this section into two parts. The first part presents the statistical and visual characteristics of the cleaned and preprocessed data. The second part presents the performance of the applied algorithms.

A. Exploratory Data Analysis (EDA)

To ensure consistent assumptions across models, we applied a standardized process of data cleaning and preprocessing.

We first cleaned the dataset by retaining only benign and DDoS traffic, eliminating irrelevant variables (e.g., identifiers), and handling infinite, missing, and invalid values.

The final number of features (including the target feature) and rows, as well as the distribution of benign and DDoS traffic across the analyzed datasets, are presented in Table II.

Table II shows that all datasets are relatively well-balanced. While this is normally desirable, it requires certain changes to the data splitting process during the development of anomaly detection models. In particular, due to the assumption of anomalies being infrequent events, their performance may not be representative if tested on balanced datasets.

We observe that features related to time, packet frequency, and volume follow a right-skewed distribution in each dataset (Fig. 5). Since the flow duration parameter has a different time scale in two of the datasets, its distribution is not shown.

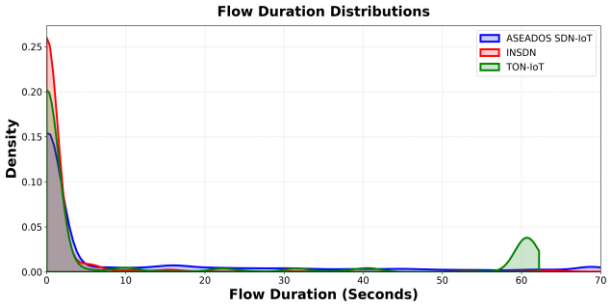


Fig. 5. Flow duration distribution across three datasets.

PCA on datasets with highly correlated columns results in a smaller number of extracted features (Fig. 6). Semi-supervised ML uses only the benign data for training. Therefore, the number of extracted features for semi-supervised learning is naturally lower compared to other unsupervised methods. An exception is the number of extracted features for the Cloud dataset, which is likely due to the diversity of benign traffic and the similarity of benign and malicious traffic. Chi-squared, which is used for supervised ML algorithms, shows the degree to which the features are correlated with the label column.

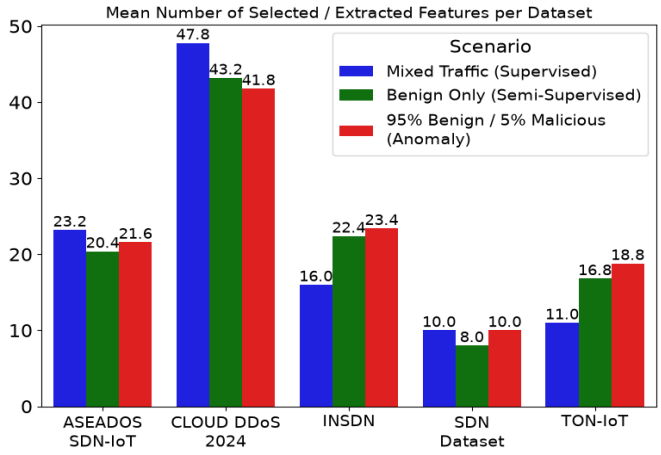


Fig. 6. Mean number of selected/extracted features across datasets.

The most important features as selected by χ^2 are presented in Appendix D (Table VII).

TABLE II. Dataset After Cleaning

Cleaned dataset	Benign	DDoS	Features	Total
ASEADOS SDN IOT	246.071 (58.15%)	177.056 (41.85%)	86	423.127
BCCC-cPacket-CloudDDoS-2024	260.205 (81.45%)	59.246 (18.55%)	300	319.451
SDN dataset	1.029.143 (65.66%)	538.220 (34.34%)	30	1.567.363
TON-IoT	410.476 (50.26%)	406.190 (49.74%)	50	816.666
INSDN	58.758 (53.20%)	51.673 (46.8%)	86	110.431

B. Model Results

Most studies in this field consider accuracy a leading classification performance metric. However, in highly imbalanced learning environments, such as those for anomaly detection, where anomalies are expected to be rare events, relying on accuracy can be misleading. This is due to the fact that accuracy is highly influenced by class distribution and may remain high even when false positives and false negatives are considerable.

To ensure fairness and consistency in our analysis, we use the F1-score as the primary metric for model performance evaluation, as it balances precision and recall. Fig. 7 presents the mean F1 scores and standard deviations of the models on each dataset.

Among the selected models, XGBoost demonstrated the best classification results, whereas SGD-OCSVM performed the worst regarding classification metrics. The small standard deviation observed for all models indicates that the classification results provide a reliable estimate of the expected model performance, suggesting good generalization across different runs and dataset splits.

Table III presents the performance of all models across all five datasets, whereas Fig. 8 illustrates the trade-off between the model classification performance and computational efficiency across all datasets. All values were obtained from an isolated dedicated cloud environment.

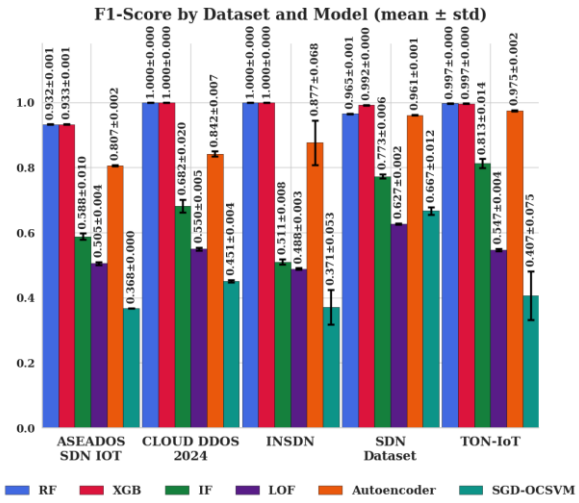


Fig. 7. F1 scores and standard deviations of models on each dataset.

As shown in Fig. 8, XGBoost achieves the best balance between DDoS identification and computational efficiency. Among unsupervised methods, IF stands out by combining solid classification performance with high computational efficiency. The autoencoder demonstrates high classification performance at the cost of higher resource utilization and runtime. SGD-OCSVM outperforms all other methods in terms of runtime. However, it ranks last in DDoS identification performance across the selected datasets and consumes considerable CPU resources.

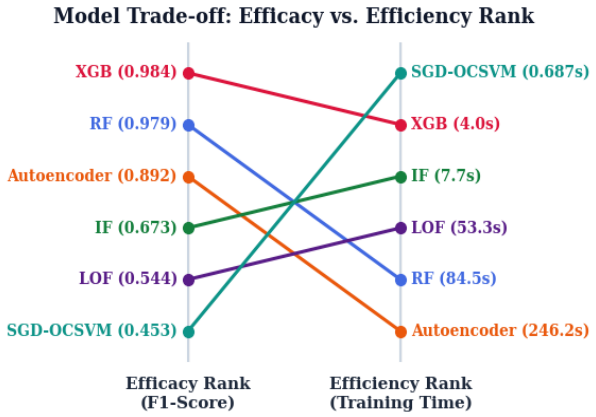


Fig. 8. Efficacy (F1) vs Efficiency (Training Time) trade-off.

Except in the case of SGD-OCSVM, where accuracy and F1 vary significantly, the low standard deviation in classification for all other models shows that they maintain high performance consistency across different networking setups. Higher variability in computational metrics across the datasets is expected, since computational efficiency highly depends on the feature space.

To assess the discriminative power of each dataset, we analyzed the mean normalized confusion matrices across datasets (Fig. 9). The TON-IoT dataset was the most discriminative compared with the other datasets, whereas ASEADOS-SDN-IoT showed less discriminative power.

TABLE III. AVERAGE CLASSIFICATION PERFORMANCE AND EFFICIENCY OF THE MODELS ACROSS FIVE DATASETS

Model/Metric	RF	XGB	IF	LOF	Autoencoder	SGD-OCSVM
Accuracy	0.980 ± 0.029	0.985 ± 0.029	0.938 ± 0.024	0.913 ± 0.010	0.911 ± 0.062	0.649 ± 0.132
Precision	0.981 ± 0.030	0.984 ± 0.030	0.673 ± 0.126	0.543 ± 0.053	0.912 ± 0.057	0.564 ± 0.238
Recall	0.978 ± 0.029	0.985 ± 0.026	0.673 ± 0.126	0.544 ± 0.054	0.885 ± 0.078	0.542 ± 0.068
F1-Score	0.979 ± 0.030	0.984 ± 0.029	0.673 ± 0.126	0.544 ± 0.054	0.892 ± 0.074	0.453 ± 0.125
Training Time (s)	84.493 ± 120.613	3.997 ± 3.042	7.696 ± 5.757	53.291 ± 40.846	246.215 ± 274.419	0.687 ± 0.466
Tuning Time (s)	5860.820 ± 7326.411	668.243 ± 373.508	175.759 ± 135.280	1081.502 ± 984.672	18138.425 ± 12948.144	16.450 ± 12.545
Training CPU (%)	25.266 ± 0.407	28.610 ± 3.922	24.998 ± 0.016	42.853 ± 10.265	34.579 ± 3.347	49.402 ± 25.011
Tuning CPU (%)	26.018 ± 0.931	32.591 ± 6.241	25.000 ± 0.000	31.620 ± 4.687	33.553 ± 2.751	66.481 ± 19.096
Training Memory Peak (MB)	69.836 ± 46.018	3.956 ± 3.507	37.491 ± 26.140	347.488 ± 415.666	659.358 ± 593.578	46.669 ± 25.931
Tuning Memory Peak (MB)	1041.916 ± 646.464	1003.888 ± 668.147	598.580 ± 504.337	721.308 ± 532.039	1349.199 ± 843.267	1163.579 ± 857.727

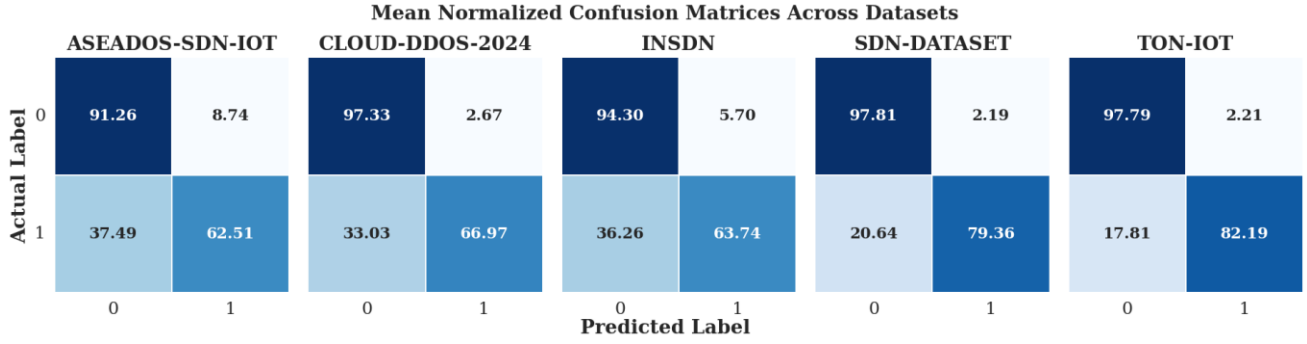


Fig. 9. Mean normalized confusion matrices for all models across each dataset.

V. Discussion

We analyzed and compared the performance of several ML approaches across five different datasets for DDoS detection. While our focus was on providing a clearer assessment of the effectiveness, efficiency, generalization, and performance consistency of shallow ML methods for different network environments, we also provided insight into the datasets used for model training.

The EDA confirmed that packet timing, frequency, and packet length metrics follow a right-skewed distribution across all datasets. We noticed shifted peak positions in the timing statistics derived from datasets containing IoT traffic compared with those from other environments. The results in Fig. 9 also indicate that SDN and TON-IoT datasets contain more discriminative features than the other datasets. The highest false-negative rates are evident in the ASEADOS-SDN-IOT dataset (37.49%) and INSDN (26.26%). This is likely due to the higher presence of slow-rate attacks in these datasets. The normalized values for the Cloud dataset also show a relatively high false-negative rate (33.03%). This is likely due to the "ambiguity" introduced by traffic labeled as "Suspicious". A more thorough analysis of datasets remains a subject for future work.

In terms of effectiveness and computational efficiency, the results indicate that XGBoost is the most balanced supervised method (F1=0.984, Training Time=4s) for DDoS detection, whereas IF performs best (F1= 0.673, Training Time=7.7s) among the evaluated unsupervised methods (Fig. 8). This result is in line with the existing literature and confirms the potential of IF for integration into defense systems in IoT environments. As IF primarily focuses on global anomalies, LOF may be used as a complementary technique to detect local anomalies (e.g., slow rate attacks).

SGD-OCSVM achieves a short runtime at the cost of relatively high CPU utilization. Autoencoders demonstrate high anomaly detection capabilities and are highly useful in many IDS systems, but their computational requirements make them less applicable to resource-constrained devices.

All models demonstrated high generalization capabilities. Apart from the F1 (0.453 ± 0.125) and accuracy (0.649 ± 0.132) scores of the OCSVM, the low standard deviations of all other classification measures indicate that the models also exhibit high performance consistency across diverse networking scenarios.

The performed analysis underscores the need for combined approaches while contributing to the selection of their components.

VI. Conclusion and Future Work

As Internet technologies become more diverse, the complexity of DDoS attacks increases, and the need for adaptive intelligent DDoS detection systems becomes more apparent. DL-based methods are superior in many tasks, but they are computationally complex for resource-constrained devices (such as IoT devices and many SDN controllers).

In this research, we developed a benchmarking framework to evaluate six prominent supervised, unsupervised, and semi-supervised learning algorithms for DDoS detection across five datasets acquired from different networking environments.

The comparative analysis highlights the capacity of simple models to balance computational efficiency and DDoS detection performance while maintaining strong generalization and consistency across the diverse network environments, emphasizing their potential for integration into hybrid systems.

Future research will use these findings to examine the adaptability of resource-efficient hybrid models to concept drift, data drift, and zero-day attacks.

References

[1] H. Olimboyev, S. Khamrayeva, H. Olimboyev, O. Khalmuratov, K. Rakhimboev and B. Ibragimov, "Using Deep Learning to Detect DDoS Attacks at the Application Layer," 2025 IEEE 26th International Conference of Young Professionals in Electron Devices and Materials (EDM), Altai, Russian Federation, 2025, pp. 2010-2016, doi: 10.1109/EDM65517.2025.11096735.

[2] Ö. Tonkal and J. A. Mgungile, "Evaluating Machine Learning Models for DDoS Detection in SDNs," J. Adv. Inf. Technol., vol. 17, no. 3, pp. 596–610, 2026, doi: 10.12720/jait.17.3.596-610.

[3] J. Pebrianto and V. Suryani, "Adaptive DDoS Attack Detection: Entropy-Based Model With Dynamic Threshold and Suspicious IP Reevaluation," IEEE Access, vol. 13, pp. 55858–55876, 2025, doi: 10.1109/ACCESS.2025.3553144.

[4] E. Cambiaso, M. Aiello, M. Mongelli, and I. Vaccari, "Detection and Classification of Slow DoS Attacks Targeting Network Servers, " *Proc. of 15th International Conf. on Availability, Reliability and Security*, Virtual event, Ireland, August, 2020, doi: 10.1145/3407023.3409198.

[5] Z. Tafa & V. Milutinovic, "Machine Learning in Congestion Control: A Survey on Selected Algorithms and a Roadmap to Their Implementation," arXiv:2112.15522. 2021.

[6] O. Ebrahem, S. Dowaji, and S. Alhammoud, "On the Limited Generalizability of a Universal Features Set for DDoS Detection Across Network Environments," IEEE Access, vol. 14, pp. 7932–7974, 2026, doi: 10.1109/ACCESS.2026.3653648.

[7] F. Luo, S. Fan, and G. Shao, "A Survey of Emerging DDoS Threats in New Power Systems," Sensors, vol. 26, no. 4, p. 1097, Jan. 2026, doi: 10.3390/s26041097

[8] M. M. Trigui, M. S. Islam, M. D. Gambo, M. S. Alam, and T. Helmy, "Resource-Aware Ensemble Framework for DDoS Detection in SDN via Novel Game-Theoretic Feature Selection," IEEE Access, vol. 14, pp. 1666–1676, 2026, doi: 10.1109/ACCESS.2025.3618519.

[9] A. Hirsi et al., "HSF: A Hybrid SVM-RF Machine Learning Framework for Dual-Plane DDoS Detection and Mitigation in Software-Defined Networks," IEEE Access, vol. 13, pp. 112303–112323, 2025, doi: 10.1109/ACCESS.2025.3583712.

[10] S. Peng, J. Tian, X. Zheng, S. Chen, and Z. Shu, "DDoS Defense Strategy Based on Blockchain and Unsupervised Learning Techniques in SDN," Future Internet, vol. 17, no. 8, p. 367, Aug. 2025, doi: 10.3390/fi17080367.

[11] S. Das, M. Ashrafuzzaman, F. T. Sheldon, and S. Shiva, "Ensembling Supervised and Unsupervised Machine Learning Algorithms for Detecting Distributed Denial of Service Attacks," Algorithms, vol. 17, no. 3, p. 99, Mar. 2024, doi: 10.3390/a17030099.

[12] M. Roopak, S. Parkinson, G. Y. Tian, Y. Ran, S. Khan, and B. Chandrasekaran, "An Unsupervised Approach for the Detection of Zero-Day DDoS Attacks in IoT Networks," Authorea, vol. 2024, no. 0114, doi: 10.22541/au.170526630.07302484/v1.

[13] N. U. Ain, M. Sardaraz, M. Tahir, M. W. Abo Elsoud, and A. Alourani, "Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach," Sensors, vol. 25, no. 5, p. 1346, Jan. 2025, doi: 10.3390/s25051346.

[14] T. Lakshan Yasarathna and N.-A. Le-Khac, "ASEADOS-SDN-IoT: A novel SDN-IoT network intrusion detection dataset and framework," Internet Things, vol. 36, p. 101891, Mar. 2026, doi: 10.1016/j.iot.2026.101891.

[15] M. Shafi, A. H. Lashkari, V. Rodriguez, and R. Nevo, "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS)

Dataset and Cloud Intrusion Traffic Characterization," Information, vol. 15, no. 4, p. 195, Apr. 2024, doi: 10.3390/info15040195.

[16] A. Kaan Sarica and P. Angin, "A Novel SDN Dataset for Intrusion Detection in IoT Networks," in 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey: IEEE, Nov. 2020, pp. 1–5. doi: 10.23919/CNSM50824.2020.9269042.

[17] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," IEEE Access, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[18] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A Novel SDN Intrusion Dataset," IEEE Access, vol. 8, pp. 165263–165284, 2020, doi: 10.1109/ACCESS.2020.3022633.

APPENDICES

TABLE IV. APPENDIX A: LIBRARY VERSIONS

Library	Version	Usage
Python	3.12.3 with GCC 13.3.0	Experimental environment
Pandas	2.3.3	Data preprocessing and analyses
NumPy	2.4.2	
matplotlib	3.10.8	Graph generation
seaborn	0.13.2	
scikit-learn	1.8.0	ML models, preprocessing, and model evaluation metrics
XGBoost	3.3.0	XGBoost model building
TensorFlow	2.21.0 (Keras API)	To build Autoencoder model
PyTorch	2.13.0	
kneed	0.8.6	For knee-point detection
psutil	7.2.2	Resource utilization monitoring.

TABLE V. APPENDIX B: AUTOENCODER ARCHITECTURE AND OPTIMIZATION

ALGORITHM: Autoencoder model creation
Input: input_dim, encoding_dim inputs ← Input(input_dim) encoded_1 ← Dense (encoding_dim, ReLU) (inputs) encoded_2 ← Dense (encoding_dim / 2, ReLU) (encoded_1) decoded_1 ← Dense (encoding_dim, ReLU) (encoded_2) outputs ← Dense (input_dim, Linear) (decoded_1) Autoencoder ← Model (inputs, outputs) Optimizer ← Adam Loss ← Mean Squared Error Compile Autoencoder using Optimizer and Loss Return Autoencoder

TABLE VI. APPENDIX C: HYPERPARAMETERS USED FOR HYPERPARAMETER TUNING

Model	Hyperparameter	Values Tested
Random Forest	n_estimators	100, 200
	max_depth	5, 10, 15
	min_samples_split	2, 5, 10
	min_samples_leaf	1, 2, 4
	max_features	sqrt
	random_state	42
	n_jobs	1
XGBoost	n_estimators	100, 200
	max_depth	3, 6, 10
	learning_rate	0.01, 0.1, 0.2
	subsample	0.8, 1.0
	colsample_bytree	0.8, 1.0
	objective	binary
	eval_metric	logloss
	tree_method	hist
	random_state	42
	n_jobs	1
Local Outlier Factor	n_neighbors	10, 20, 30
	algorithm	auto,ball_tree, kd_tree
	Contamination	0.05
	novelty	True
Isolation Forest	n_estimators	100, 200, 300
	max_samples	auto, 0.8, 0.5
	contamination	0.05
	random_state	42
	n_jobs	1
SGD One-Class SVM	eta0	0.001, 0.01
	nu	0.05, 0.1, 0.2
	Learning_rate	optimal
	average	True
	random_state	42
Autoencoder	encoding_dim	8, 12,16, 24, 32, 48
	epochs	30, 50
	batch_size	128, 256
	threshold_percentile	95, 97, 99

TABLE VII. APPENDIX D: FEATURES SELECTED ACROSS 5 FOLDS

Dataset	Nr. of Features	Selected features
CLOUD-DDOS-2024	47	max_header_bytes, min_header_bytes, mean_header_bytes, median_header_bytes, mode_header_bytes, bwd_std_header_bytes, bwd_cov_header_bytes, bwd_init_win_bytes, syn_flag_percentage_in_total, fwd_syn_flag_percentage_in_total, fwd_ack_flag_percentage_in_total, bwd_syn_flag_percentage_in_total, fwd_syn_flag_percentage_in_fwd_packets, fwd_ack_flag_percentage_in_fwd_packets, bwd_syn_flag_percentage_in_bwd_packets, bwd_ack_flag_perczentage_in_bwd_packets, bwd_rst_flag_percentage_in_bwd_packets, packets_IAT_mean, packet_IAT_max, packet_IAT_min, packet_IAT_total, packets_IAT_median, packets_IAT_cov, packets_IAT_mode, fwd_packets_IAT_cov, bwd_packets_IAT_mean, bwd_packets_IAT_max, bwd_packets_IAT_min,,

		bwd_packets_IAT_total, bwd_packets_IAT_median, bwd_packets_IAT_mode, handshake_state, min_bwd_packets_delta_time, max_bwd_packets_delta_time, mean_packets_delta_time, mode_packets_delta_time, median_packets_delta_time, cov_packets_delta_time, mean_bwd_packets_delta_time, mode_bwd_packets_delta_time, median_bwd_packets_delta_time, cov_fwd_packets_delta_time, std_header_bytes_delta_len, activity, Src_Port_Freq, Src_Port_Group_Private, Src_Port_Group_Registered
TON-IOT	11	ts, conn_state, dns_qtype, dns_AA, dns_RA, Src_Port_Freq, Dst_Port_Freq, Src_Port_Group_Well-known, Dst_Port_Group_Private, Dst_Port_Group_Registered, Dst_Port_Group_Well-known
SDN-DATASET	10	Pkts, Bytes, Srate, Drate, TnP_PSrcIP, TnP_PDstIP, TnP_PerProto, TnP_Per_Dport, N_IN_Conn_P_DstIP, N_IN_Conn_P_SrcIP
ASEADOS-SDN-IOT	23	Flow Duration, Fwd Pkt Len Max, Flow IAT Max, Fwd IAT Tot, Fwd IAT Std, Fwd IAT Max, Bwd IAT Tot, Bwd IAT Std, Bwd IAT Max, Bwd PSH Flags, FIN Flag Cnt, SYN Flag Cnt, RST Flag Cnt, PSH Flag Cnt, Idle Mean, Idle Max, Idle Min, Src_Port_Freq, Src_Port_Group_Unknown, Dst_Port_Group_Private, Dst_Port_Group_Registered, Dst_Port_Group_Unknown, Dst_Port_Group_Well-known
INSDN	16	Protocol, Flow Duration, Bwd Pkt Len Min, Fwd IAT Tot, Fwd IAT Std, Fwd IAT Max, Bwd IAT Tot, Pkt Len Min, FIN Flag Cnt, SYN Flag Cnt, ACK Flag Cnt, Src_Port_Freq, Dst_Port_Freq, Src_Port_Group_Registered, Src_Port_Group_Well-known, Dst_Port_Group_Private

A Confidence-Calibrated Adaptive Ensemble Voting Framework for Stock Price Direction Prediction

Yuan Lukito*, Nugroho Agus Haryono, Raden Gunawan Santosa
Department of Informatics, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia

***Abstract*—Predicting stock price direction is crucial for investment decisions, yet most machine learning models optimize classification metrics while ignoring probability calibration. Uncalibrated probabilities often produce overconfident predictions, increasing financial risk. To address this gap, we introduce Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C), a framework combining temperature scaling calibration with dynamic confidence-weighted aggregation. We evaluate CAEV-C using XGBoost, LightGBM, and Bidirectional Long Short-Term Memory (BiLSTM) base learners across seven technology stocks (AAPL, AMZN, GOOGL, META, MSFT, NVDA, and TSLA) under a temporal walk-forward validation scheme. CAEV-C achieves the highest average F1-score of 0.5885, outperforming LightGBM (0.4399), Hard Voting (0.5391), and Soft Voting (0.4451). Furthermore, CAEV-C yields the lowest Expected Calibration Error (ECE) of 0.0347, improving calibration quality over LightGBM (0.2053), Hard Voting (0.2629), and Soft Voting (0.1062). Reliability diagrams confirm that CAEV-C confidence estimates track ideal calibration closely, while Wilcoxon Signed-Rank Tests demonstrate statistically significant ECE improvements over Hard and Soft Voting, as well as significant F1-score gains over Soft Voting ($p < 0.05$). By improving probability reliability while maintaining competitive predictive performance, CAEV-C provides a robust ensemble framework for quantitative stock forecasting.**

***Keywords*—Stock market prediction; ensemble learning; confidence-calibrated ensemble; probability calibration**

I. Introduction

Stock price direction prediction helps investors, financial institutions, and researchers guide trading decisions, portfolio optimization, and risk management. However, forecasting market movements remains difficult because financial time series are noisy, non-stationary, and nonlinear [1]. Macroeconomic conditions, investor sentiment, and unexpected events continuously alter market dynamics, limiting the effectiveness of traditional analytical models. These challenges drive ongoing research into computational methods that handle complex financial time-series data more reliably.

To overcome these limits, researchers frequently turn to machine learning and deep learning algorithms [2]. Gradient boosting models like eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM) capture complex nonlinear patterns in financial features [3], whereas recurrent architectures such as Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) process sequential market dependencies [4]. However, single models perform unevenly across different stocks and market regimes. This inconsistency reduces generalization, leading researchers toward ensemble learning methods that combine diverse models.

Ensemble learning improves prediction robustness by combining outputs from multiple models to reduce individual biases. Standard aggregation techniques include Hard Voting, which takes majority decisions, and Soft Voting, which averages predicted probabilities to incorporate confidence scores [5]. Yet conventional voting mechanisms assume these predicted probabilities are reliable. In practice, machine learning and deep learning models produce uncalibrated probabilities, yielding overconfident or underconfident estimates. Poorly calibrated confidence scores distort ensemble decisions [6] by hiding actual market uncertainty, making probability-weighted voting less reliable.

Stock forecasting studies typically evaluate models using classification metrics such as Accuracy, F1-score, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) [1]. While these metrics measure discrimination ability, they do not show whether predicted probabilities reflect actual outcomes. A model can achieve strong classification scores while producing poorly calibrated probabilities, yielding overconfident or underconfident estimates [7]. This gap matters in finance, where trading decisions rely on both predicted direction and prediction confidence. Calibration metrics like Expected Calibration Error (ECE) and Brier Score measure how well probabilities align with real outcomes. However, calibration quality remains rarely evaluated in stock direction research, particularly within ensemble learning frameworks.

Although ensemble learning is widely used in stock market forecasting [5], researchers rarely evaluate the reliability of the confidence estimates used during aggregation. Most studies combine model outputs via Hard or Soft Voting without first calibrating prediction probabilities. Consequently, uncalibrated base predictions carry over into the ensemble output, degrading decision reliability. Most stock market forecasting studies emphasize classification accuracy while overlooking calibration quality. Addressing this gap requires an ensemble framework that combines predictions from multiple models while calibrating probabilities to match real-world outcomes.

To address these limitations, we propose Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) for stock price direction prediction. The framework integrates predictions from three diverse base models: XGBoost, LightGBM, and BiLSTM. Before ensemble aggregation,

*Corresponding author.

temperature scaling calibrates each model's probability outputs to reduce overconfidence and improve reliability. The framework then uses these calibrated probabilities in a confidence-weighted aggregation scheme, weighting predictions by their confidence levels. By combining calibration with confidence-weighted voting, CAEV-C improves probability reliability while maintaining competitive predictive performance for financial decision-making.

We evaluate CAEV-C using walk-forward validation across seven technology stocks: AAPL, AMZN, GOOGL, META, MSFT, NVDA, and TSLA. CAEV-C achieves the highest average F1-score of 0.5885, outperforming Hard Voting (0.5391) and Soft Voting (0.4451). It also records the lowest Expected Calibration Error (ECE) of 0.0347, improving calibration over Hard Voting (0.2629) and Soft Voting (0.1062). Wilcoxon Signed-Rank Tests confirm statistically significant ECE gains over both baselines and significant F1-score improvements over Soft Voting.

This study offers three main contributions: 1) a confidence-calibrated ensemble voting framework (CAEV-C) for stock price direction prediction, 2) A pipeline combining temperature scaling with confidence-weighted aggregation to calibrate probabilities before decision-making, 3) An empirical evaluation across multiple stocks, calibration metrics, and statistical significance tests to validate predictive performance and reliability.

II. Related Work

A. Stock Price Direction Prediction Using Machine Learning and Deep Learning

Broader access to financial data and faster computing have driven the use of machine learning for predicting stock price directions. Unlike traditional statistical models that rely on strict assumptions about data distributions and linearity, machine learning algorithms can capture complex nonlinear patterns from large datasets [8], [9], [10]. Tree-based methods like Random Forest [11], XGBoost [3], [12], and LightGBM [13], [14] are widely used in stock forecasting for their strong predictive performance, robustness, and ability to handle varied feature types.

Beyond traditional machine learning, deep learning models are widely used for financial time-series forecasting. Recurrent neural networks, specifically LSTM [15], [16], [17], [18] and BiLSTM [19], help capture time-dependent patterns in historical market data. These architectures overcome standard neural networks' difficulty with long sequences, allowing them to track shifting market trends over time. Consequently, deep learning provides a strong alternative for stock direction prediction, particularly when price movements rely on complex sequential patterns.

While machine learning and deep learning models perform well, individual model accuracy varies across different stocks, market conditions, and time horizons. Tree-based models excel at finding feature interactions, whereas deep learning architectures are better at tracking temporal patterns. Because no single model outperforms all others in every scenario, combining predictors through ensemble learning allows models to complement each other, improving overall forecast stability and generalization.

B. Ensemble Learning for Stock Market Prediction

Ensemble learning is widely used in stock market forecasting to address the limits of individual models [20], [21], [22], [23], [24]. By combining predictions from multiple algorithms, ensemble methods leverage each model's strengths to offset individual weaknesses. Merging these outputs lowers prediction variance and improves overall stability and generalization. Given the high uncertainty and shifting conditions in financial markets, ensemble approaches tend to yield more reliable results than single models across different tasks and environments.

Hard Voting and Soft Voting are two of the most common aggregation techniques in ensemble learning [5]. Hard Voting bases the final prediction on a majority vote across classifiers, which is straightforward to implement and resilient to isolated model errors. Soft Voting instead averages predicted probabilities from each model and selects the class with the highest combined score. By incorporating probability estimates rather than just binary votes, Soft Voting provides more nuanced decisions than simple majority voting. Consequently, financial forecasting research frequently uses both techniques as standard baselines.

Despite their usefulness, traditional voting ensembles have clear drawbacks. Hard Voting relies entirely on class labels and ignores prediction confidence, discarding helpful probability data. Soft Voting assumes that a model's predicted probabilities are accurate and well calibrated. In practice, machine learning and deep learning models often produce overconfident or underconfident estimates [6], which skews aggregated probabilities away from the true likelihood of outcomes. Poor calibration can undermine Soft Voting, pointing to the need for ensemble methods that directly factor in probability reliability when merging predictions.

C. Confidence Calibration and Probability Reliability

Although high accuracy is a primary goal in stock market forecasting, accurate classification alone does not guarantee trustworthy probability estimates. A model can achieve a high Accuracy, F1-score, or ROC-AUC while outputting probabilities that misrepresent the true likelihood of an outcome. For instance, if a model assigns 90% confidence to a set of predictions, a properly calibrated system should be right about 90% of the time. In practice, machine learning and deep learning models often display overconfidence or underconfidence, creating a gap between predicted probabilities and actual outcomes. Because financial decisions rely on both predicted directions and confidence levels, miscalibration can skew risk assessments and lead to poor choices.

Researchers use several calibration techniques to make probabilistic predictions more reliable, including Platt Scaling, Isotonic Regression, and Temperature Scaling [25]. Temperature Scaling is especially popular because it is simple and effective. It introduces a single temperature parameter to adjust output confidence without changing prediction rankings, keeping classification decisions intact while improving

probability estimates. Because it is computationally efficient and performs well in practice, Temperature Scaling is widely used as a post-processing step for machine learning and deep learning models.

Several metrics help assess probability calibration quality [26]. Expected Calibration Error (ECE) measures the gap between predicted confidence and actual accuracy across confidence bins. The Brier Score calculates the mean squared error between predicted probabilities and actual outcomes to evaluate probability accuracy. Researchers also use reliability diagrams to visually compare prediction confidence against empirical accuracy. Together, these tools provide complementary views on whether a model's probabilities are trustworthy for real-world decision-making.

Despite the importance of confidence reliability, few stock forecasting studies evaluate model calibration. Most research focuses on standard classification metrics like Accuracy, F1-score, and ROC-AUC, paying little attention to whether predicted probabilities are trustworthy. Furthermore, while ensemble learning is widely used to boost forecasting accuracy, combining probability calibration directly with ensemble aggregation remains rare. This gap creates an opportunity for ensemble frameworks that incorporate confidence calibration, delivering reliable probability estimates alongside strong predictive performance.

D. Research Gap and Research Positioning

Recent research has advanced stock price direction forecasting using machine learning, deep learning, and ensemble techniques. Machine learning algorithms like XGBoost and LightGBM fit nonlinear patterns in financial data, while deep learning architectures such as LSTM and BiLSTM effectively model temporal dependencies. Ensemble methods, particularly Hard Voting and Soft Voting, combine multiple models to improve prediction stability. However, existing studies focus primarily on standard classification metrics like Accuracy, F1-score, and ROC-AUC, paying far less attention to the reliability of predicted probabilities. Consequently, confidence calibration remains largely underexplored in stock market prediction.

While probability calibration is common in general machine learning, its use in ensemble-based stock prediction remains rare. Most ensemble methods aggregate model outputs without calibrating probabilities before voting, which can pass unreliable confidence estimates into final predictions. This study addresses that issue by introducing the Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) framework. CAEV-C pairs temperature scaling calibration with confidence-aware ensemble aggregation. Unlike standard voting approaches that focus strictly on accuracy, CAEV-C evaluates both predictive performance and confidence reliability to deliver reliable forecasts for financial decision-making.

CAEV-C combines temperature scaling probability calibration with confidence-aware ensemble aggregation, allowing the system to weigh model predictions alongside their confidence reliability. By evaluating accuracy and calibration quality together through the F1-score, Brier Score, and Expected Calibration Error (ECE), the framework provides a more reliable ensemble forecasting approach for financial decisions.

III. Methodology

A. Proposed CAEV-C Framework

This study introduces the Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) framework for stock price direction prediction. By unifying diverse models, probability calibration, and confidence-aware voting into a single pipeline, the framework improves both forecasting accuracy and probability reliability. Unlike conventional ensembles that combine raw model outputs directly, CAEV-C calibrates each model's probability estimates before voting. This step reduces the impact of overconfident or underconfident predictions, leading to more reliable forecasts.

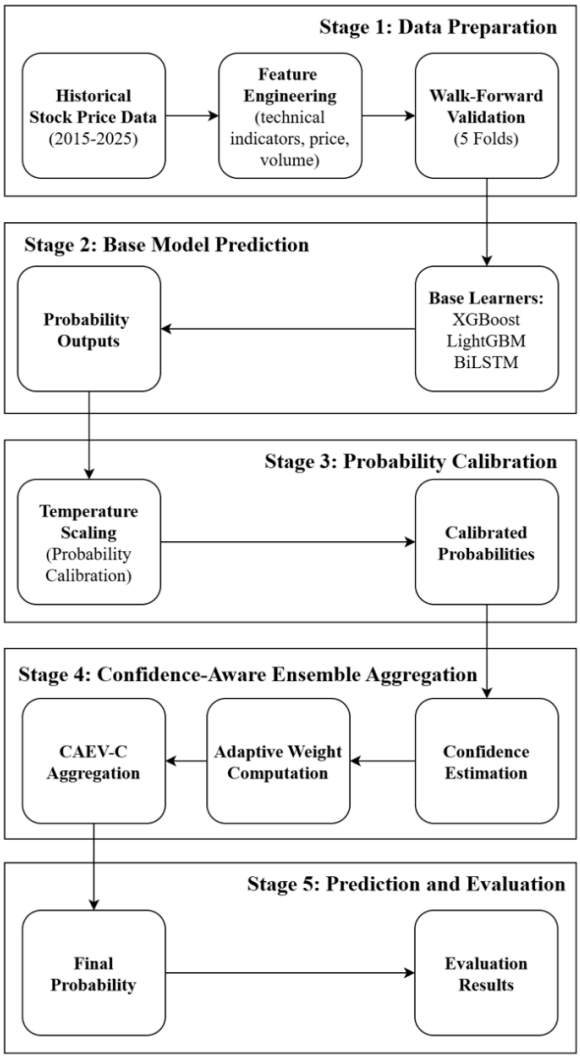


Fig. 1. Overall architecture of the proposed CAEV-C framework consisting of a five-stage workflow.

The framework follows a five-stage workflow as depicted in Fig. 1. First, historical market data is transformed into predictive features and split using walk-forward validation to

preserve temporal order. Second, three distinct base models (XGBoost, LightGBM, and BiLSTM) are trained independently to capture different patterns in financial time series. Third, Temperature Scaling calibrates each model's probability outputs to align predicted confidence with observed outcomes. Fourth, these calibrated probabilities determine prediction confidence levels and calculate adaptive voting weights. Finally, the framework aggregates the calibrated probabilities through confidence-aware voting to produce the final stock direction forecast.

Combining probability calibration with adaptive weighting allows the ensemble to evaluate both predicted class labels and the reliability of their confidence scores. As a result, CAEV-C balances classification accuracy and probability reliability better than standard Hard Voting or Soft Voting.

B. Base Prediction Models

The CAEV-C framework uses three distinct base learners (XGBoost, LightGBM, and BiLSTM) to capture different characteristics of stock market data. Combining diverse models increases prediction diversity, a key factor in ensemble performance. Tree-based algorithms excel at fitting nonlinear relationships in engineered financial features, while recurrent neural networks capture temporal patterns in sequential market data. Merging these complementary approaches provides the ensemble with varied predictive perspectives before calibration and aggregation.

1) XGBoost: eXtreme Gradient Boosting (XGBoost) is a gradient-boosted decision tree algorithm that sequentially constructs trees to minimize a specified loss function. Each new tree corrects the residual errors of earlier trees, allowing the model to learn complex nonlinear relationships in data. Because of its built-in regularization and efficient optimization, XGBoost is widely used in stock market prediction and financial forecasting.

For a dataset with N observations and F* representing the number of features, the XGBoost prediction is given by Eq. (1), where $f_k \in F$, denotes the *k-th* decision tree, F represents the space of regression trees, and K represents the total number of trees in the ensemble.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (1)$$

For binary stock direction prediction, the model outputs the estimated probability of an upward price movement, which is then used in the calibration and ensemble aggregation stages.

2) LightGBM: Light Gradient Boosting Machine (LightGBM) is a gradient boosting framework built for high speed and scalability without sacrificing predictive accuracy. Unlike traditional tree-based algorithms that grow trees level by level, LightGBM uses a leaf-wise growth strategy, expanding whichever leaf offers the highest information gain. This allows the model to achieve high accuracy with fewer trees.

For a given instance x_i, the LightGBM prediction is expressed as in Eq. (2), where g_k denotes the *k-th* boosted tree in the LightGBM ensemble.

$$\hat{y}_i = \sum_{k=1}^{K} g_k(x_i) \quad (2)$$

LightGBM efficiently learns feature interactions, making it well-suited for stock market prediction, where relationships between technical indicators and market variables are often nonlinear and dynamic. As with XGBoost, LightGBM's probability outputs are calibrated before being integrated into the ensemble framework.

3) Bidirectional Long Short-Term Memory (BiLSTM): Bidirectional Long Short-Term Memory (BiLSTM) extends the standard LSTM architecture by processing sequential data in both forward and backward directions. By combining past and future context within a sequence, BiLSTM captures richer temporal representations than unidirectional LSTM networks.

Given an input sequence as expressed in Eq. (3):

$$X = \{x_1, x_2, \ldots, x_T\} \quad (3)$$

The forward and backward hidden states are computed as in Eq. (4) and Eq. (5):

$$\overrightarrow{h_t} = LSTM_{forward}\left(x_t, \overrightarrow{h_{t-1}}\right) \quad (4)$$

$$\overleftarrow{h_t} = LSTM_{backward}\left(x_t, \overleftarrow{h_{t+1}}\right) \quad (5)$$

The final hidden representation is obtained by concatenating both directions as expressed in Eq. (6), where [;] denotes the concatenation operation.

$$h_t = \left[\overrightarrow{h_t}\,;\, \overleftarrow{h_t}\right] \quad (6)$$

In stock price direction prediction, BiLSTM captures temporal dependencies and sequential patterns that tree-based models may miss. Its output layer estimates the probability of an upward price movement, which is then calibrated and integrated into the CAEV-C aggregation process.

The predicted probabilities from these three base learners serve as the inputs for the calibration stage. To improve confidence reliability before ensemble aggregation, the framework calibrates these outputs using Temperature Scaling, as detailed in the following subsection.

C. Probability Calibration Using Temperature Scaling

Probabilities from machine learning and deep learning models often serve as confidence scores, but high accuracy does not guarantee well-calibrated probabilities. Models frequently yield overconfident probabilities that exceed true event frequencies, or underconfident ones that are overly conservative. Such calibration errors pose risks in stock forecasting because investment decisions depend on predicted market directions alongside their confidence levels. Improving probability reliability is therefore an essential step before probability-based ensemble aggregation.

To address this issue, the framework uses Temperature Scaling, a post-hoc calibration technique that adjusts model confidence without changing prediction rankings. Unlike calibration methods that require model retraining, Temperature Scaling introduces a single scalar parameter$T$ to rescale output logits. This approach is computationally efficient, reduces

prediction overconfidence, and preserves the original model's discriminative ability.

If z represents a classifier's logit before probability transformation, Temperature Scaling computes the calibrated probability as expressed in Eq. (7), where $\hat{p}$ denotes the calibrated probability, z represents the original logit output, T denotes the temperature parameter and $\sigma(.)$ is the sigmoid activation function.

$$\hat{p} = \sigma\left(\frac{z}{T}\right) \quad (7)$$

The temperature parameter controls the sharpness of the probability distribution. At $T = 1$, probability estimates remain unchanged. Setting $T > 1$ softens the distribution, reducing overconfidence, while $T < 1$ sharpens it, increasing prediction confidence.

The framework optimizes an independent temperature parameter for each base learner (XGBoost, LightGBM, and BiLSTM). For each walk-forward fold, the optimal temperature is determined on the validation set by minimizing Negative Log-Likelihood (NLL). Given a validation dataset D_{val} with N samples, the optimization objective is defined as in Eq. (8).

$$T^* = arg\ min_T \left(-\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{p}_i) + (1 - y_i)\ \log(1 - \hat{p}_i)]\right) \quad (8)$$

T^* denotes the optimal temperature parameter, y_i represents the ground-truth class label and $\hat{p}_i$ is the calibrated probability of sample i.

Once the optimal temperature parameters are set, the calibrated probabilities serve as inputs to the confidence estimation stage. Reducing calibration errors before ensemble aggregation ensures that the confidence values used during voting accurately reflect each model's true reliability. This calibrated confidence information provides the foundation for the CAEV-C mechanism, detailed in the following subsection.

D. Confidence Estimation

After probability calibration, the framework estimates the confidence of each model prediction. This step forms the foundation of CAEV-C's adaptive aggregation mechanism, quantifying individual prediction reliability before combining outputs. Unlike conventional voting approaches that treat all predictions equally, CAEV-C assigns greater weight to predictions with higher calibrated confidence, allowing the ensemble to adapt to each model's reliability.

Let $\hat{P}_m$ denote the calibrated probability generated by model m, where $\hat{P}_m \in [0, 1]$ and $m \in \{XGBoost, LightGBM, BiLSTM\}$. The confidence score is the maximum probability assigned to either class as expressed in Eq. (9):

$$c_m = \max\left(\hat{P}_m, 1 - \hat{P}_m\right) \quad (9)$$

where, c_m denotes the confidence score of m, and $\hat{P}_m$ represents the calibrated probability of the positive class. This yields confidence values within the interval $0.5 \leq c_m \leq 1.0$.

A confidence value near 0.5 reflects high uncertainty because both classes have nearly equal probabilities. Conversely, a value approaching 1.0 shows that the model strongly favours one class over the other. Because confidence scores come from calibrated probabilities rather than raw model outputs, they more accurately reflect the true reliability of individual predictions. This reliability is vital in financial forecasting, where prediction confidence directly influences investment decisions.

The estimated confidence scores are then converted into adaptive voting weights. Models with higher calibrated confidence carry greater weight during ensemble aggregation, while lower-confidence models contribute proportionally less to the final forecast. By incorporating calibrated confidence into the voting process, the framework reduces the impact of unreliable predictions and improves overall ensemble robustness.

The confidence scores from this stage are used in the next subsection to calculate adaptive voting weights and construct the CAEV-C mechanism.

E. Proposed Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C)

This subsection details the Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) framework. CAEV-C combines calibrated probability estimates from multiple heterogeneous base learners, adjusting each model's contribution according to its calibrated confidence level. Unlike Hard Voting, which considers only class labels, and Soft Voting, which averages probabilities without considering confidence reliability, CAEV-C integrates probability calibration and confidence-aware weighting into a single decision-making process.

The goal of CAEV-C is to improve both predictive accuracy and probability reliability. Giving greater weight to models with higher calibrated confidence reduces the influence of less reliable predictions on the final ensemble output.

1) Calibrated probability generation: After training, the three base learners (XGBoost, LightGBM, and BiLSTM) each output a probability estimate for the positive class: $P_{xgboost}, P_{lightgbm}, P_{bilstm}$. These estimates are then calibrated using the Temperature Scaling procedure from the previous subsection, yielding calibrated probabilities: $\hat{P}_{xgboost}, \hat{P}_{lightgbm}, \hat{P}_{bilstm}$. This calibration aligns predicted confidence more closely with observed outcomes before the probabilities enter the ensemble aggregation stage.

2) Confidence-aware weight computation: For each calibrated probability, the confidence score is calculated as given in Eq. (9). The confidence scores are then normalized to generate adaptive voting weights as expressed in Eq. (10):

$$w_m = \frac{c_m}{\sum_{j=1}^{M} c_j} \quad (10)$$

where, w_m denotes the adaptive weight assigned to model m and M represents the total number of base learners. This normalization process ensures that the sum of the weights across all M components sums to 1. Consequently, models

with higher calibrated confidence receive larger voting weights and carry more influence in the final ensemble prediction.

3) Confidence-calibrated ensemble aggregation: The ensemble probability is calculated through a weighted combination of calibrated probabilities as expressed in Eq. (11).

$$P_{CAEV-C} = \sum_{m=1}^{M} w_m \hat{P}_m \quad (11)$$

For the three-model ensemble used in this study, the aggregation can be expressed as Eq. (12), where the sum of the adaptive weight is equal to 1, as expressed in Eq. (12).

$$w_{xgboost} + w_{lightgbm} + w_{bilstm} = 1 \quad (12)$$

This formulation allows the final ensemble prediction to account for both predicted probabilities and their underlying reliability.

4) Final decision rule: The final stock price direction is set using a binary classification threshold as expressed in Eq. (13), where $\hat{y} = 1$ indicates an upward stock price movement and $\hat{y} = 0$ indicates a downward stock price movement.

$$\hat{y} = \begin{cases} 1, P_{CAEV-C} \geq 0.5 \\ 0, P_{CAEV-C} < 0.5 \end{cases} \quad (13)$$

5) Algorithm of CAEV-C: The CAEV-C framework operates through the following steps as expressed in Algorithm 1.

Algorithm 1: CAEV-C

Input:
Historical stock features (X)
Base learners: XGBoost, LightGBM, BiLSTM
Output:
$\hat{y}$

Train each base learner independently

Generate probability predictions: $P_{xgboost}, P_{lightgbm}, P_{bilstm}$

Apply Temperature Scaling to obtain: $\hat{P}_{xgboost}, \hat{P}_{lightgbm}, \hat{P}_{bilstm}$

Compute calibrated confidence scores: $c_m = \max(\hat{P}_m, 1 - \hat{P}_m)$

Normalize confidence scores: $w_m = \frac{c_m}{\sum_{j=1}^{M} c_j}$

Aggregate calibrated probabilities: $P_{CAEV-C} = \sum_{m=1}^{M} w_m \hat{P}_m$

Generate final prediction: if $P_{CAEV-C} \geq 0.5$
    $\hat{y}_i = 1$
else:
    $\hat{y}_i = 0$

Return $\hat{y}_i$

IV. Experimental Setup

A. Dataset Description

This study uses historical stock market data from seven major NASDAQ technology companies: Apple (AAPL), Amazon (AMZN), Alphabet (GOOGL), Meta Platforms (META), Microsoft (MSFT), NVIDIA (NVDA), and Tesla (TSLA). These stocks were selected for their heavy trading volume, large market capitalization, and diverse price behaviors, offering a reliable test environment for evaluating model robustness and generalizability. The historical market data were obtained from Yahoo! Finance, as specified in Table I.

TABLE I. Historical Market Data Obtained from Yahoo! Finance

Parameter	Description/Specification
Ticker symbols	AAPL, AMZN, GOOGL, META, MSFT, NVDA, TSLA
Observation Period	January 1st, 2015 – December 31st, 2025
Data Frequency	Daily
Data Attributes	Date, Open, High, Low, Close

The dataset contains daily trading information for each stock: open, high, low, close, adjusted close, and volume. These variables serve as inputs for feature engineering and forecasting. To ensure experimental consistency, the same preprocessing and feature generation steps were applied to every stock.

The forecasting task is framed as a binary classification problem to predict next-day stock price movements. Following standard practice in financial forecasting, target labels reflect changes in closing prices between consecutive trading days. An upward movement is assigned class 1, while a downward or flat movement is assigned class 0. The target variable is expressed as in Eq. (14), where $Close_t$ and $Close_{t+1}$ represents the closing price at trading day t and t+1, respectively.

$$y_t = \begin{cases} 1, Close_{t+1} > Close_t \\ 0, otherwise \end{cases} \quad (14)$$

Framing forecasting as a directional prediction task aligns better with real-world trading decisions than predicting exact prices. The resulting labeled dataset is then used for feature engineering, model training, probability calibration, and ensemble evaluation in the CAEV-C framework.

B. Feature Construction

Before model training, feature engineering converts raw stock market data into predictive variables. Because raw prices alone do not fully capture market conditions or investor behavior, this study uses eight technical indicators derived from historical stock data to track price returns, momentum, trend strength, and volatility.

Each of the eight indicators, as presented in Table II, highlights a specific market characteristic. Return measures relative price changes between consecutive trading periods. Momentum tracks the strength and persistence of price trends, while Moving Average Convergence Divergence (MACD) signals trend direction and momentum shifts. Average True Range (ATR) and Bollinger Bands measure market volatility. Finally, the Commodity Channel Index (CCI), Average Directional Index (ADX), and Stochastic Oscillator identify overbought or oversold conditions, trend strength, and momentum behavior.

TABLE II. TECHNICAL INDICATORS USED IN THIS STUDY

Category	Indicator	Parameters
Price Movement	Return	1 day
Momentum	Momentum Stochastic Oscillator	14 days 14 days
Trend and Trend Strength	ADX MACD	14 days 12 days (fast), 26 days (slow), 9 days (signal)
Market Condition	CCI	20 days
Volatility	ATR Bollinger Bands	14 days, 20 days (period), 2 (standard deviation)

Each technical indicator provides complementary information about market behavior. Momentum indicators capture the speed of price movements, trend indicators identify market direction, and volatility indicators measure market uncertainty. Combining these indicators gives both tree-based machine learning models and sequence-based deep learning architectures a solid representation of market activity.

All rolling technical indicators were calculated using backward-looking windows over historical price data. Feature values at time t were derived exclusively from observations up to and including trading day t, preventing future market information from contaminating feature construction.

Once feature construction is complete, the feature matrix serves as the input for XGBoost, LightGBM, and BiLSTM. The dataset is then split using a walk-forward validation strategy to preserve the temporal order of financial time-series data and prevent data leakage between training and testing periods.

C. Walk-Forward Validation Strategy

To prevent data leakage and ensure realistic model evaluation, this study uses a walk-forward validation strategy. Unlike random splitting, walk-forward validation preserves the chronological order of financial data by training models on past observations and evaluating them on future data. For each fold, models were trained on historical data, calibrated on a validation set, and evaluated on an unseen test set. After each iteration, the training window moved forward to include more recent observations. Table III presents the configuration of the five walk-forward folds.

TABLE III. THE CONFIGURATION OF THE FIVE WALK-FORWARD FOLDS

Folds	Train (Year)	Validation (Year)	Test (Year)
1.	2015 – 2019	2020	2021
2.	2015 – 2020	2021	2022
3.	2015 – 2021	2022	2023
4.	2015 – 2022	2023	2024
5.	2015 – 2023	2024	2025

To eliminate lookahead bias and prevent data leakage, training, validation, and test sets were partitioned in strict chronological sequence within each walk-forward fold. Feature scaling parameters, technical indicators, base-model training, and temperature calibration were executed strictly on data preceding the corresponding test period, ensuring no future information influenced any stage of model estimation or calibration.

The validation set was further utilized to determine the optimal temperature parameter for each base learner during the probability calibration process. Performance metrics from all testing partitions were subsequently aggregated to obtain the final evaluation results. This strategy provides a realistic assessment of stock price direction prediction models while avoiding look-ahead bias and preserving temporal consistency.

D. Experimental Environment and Hyperparameter Settings

All experiments were run in Python using Scikit-learn, XGBoost, LightGBM, and PyTorch. Pandas and NumPy handled data preprocessing and feature engineering, while custom scripts managed model evaluation and calibration analysis. Computations ran on Apple Silicon hardware using Metal Performance Shaders (MPS) acceleration for deep learning training. Maintaining a single experimental environment ensures consistency and reproducibility across all models and ensemble configurations.

1) Base model parameters: To ensure fair comparison, every model shared the same walk-forward validation protocol, feature set, and evaluation procedures. Hyperparameters as presented in Table IV were selected during preliminary experiments on the validation set and kept fixed across all walk-forward folds. Because this study focuses specifically on probability calibration and confidence-aware ensemble aggregation, extensive hyperparameter tuning was intentionally avoided to prevent confounding performance variations.

TABLE IV. HYPERPARAMETER SETTINGS

Model	Hyperparameter
XGBoost	n_estimators = 100, max_depth = 6, learning rate (eta) = 0.3, booster = gbtree subsample = 1, colsample_bytree = 1, objective = 'binary: logistic', random_state = 42
LightGBM	n_estimators = 100, num_leaves = 31, max_depth = -1, boosting_type = 'gbdt' learning_rate = 0.1, objective = binary
BiLSTM	hidden_dim = 64, num_layers = 2 dropout = 0.2, bidirectional = True fully_connected_layer = 2, lookback = 10 epochs = 60, learning_rate = 0.003 batch_size = 32, optimizer = AdamW weight_decay = 0.0001

BiLSTM input sequences were formed using a fixed lookback window of 10 trading days, excluding the boundary samples that lacked a complete historical sequence. Across all walk-forward folds, base models were initialized from scratch and trained independently, ensuring that weights and states never carried between temporal splits. To ensure reproducibility, we trained BiLSTM models with a fixed random seed (42). We applied early stopping based on validation loss with a patience of 20 epochs to prevent overfitting, and computed class weights directly from each fold's training partition to account for class imbalance.

2) Temperature scaling configuration: For probability calibration, an independent temperature parameter was optimized for each base learner within every walk-forward fold. The optimal temperature was selected on the validation

set by minimizing negative log-likelihood loss. To prevent extreme calibration behavior, the temperature search space was restricted to $0.01 \leq T \leq 10.00$. The resulting calibrated probabilities were then used by the CAEV-C framework during confidence estimation and adaptive ensemble aggregation.

Temperature parameters were estimated exclusively on the validation split of each walk-forward fold and subsequently applied to calibrate test predictions, ensuring zero test-set exposure during optimization.

These model configurations and calibration steps form the experimental setup. To evaluate CAEV-C, the following subsection introduces the baseline methods used for comparison.

E. Baseline Methods

To evaluate CAEV-C, several baseline models representing different forecasting paradigms were selected for comparison. These include individual machine learning and deep learning models, alongside traditional ensemble approaches. This experimental setup tests whether the confidence-calibrated ensemble mechanism improves upon standalone predictors and standard voting-based ensembles.

The evaluated methods fall into three groups: individual predictive models (XGBoost, LightGBM, and BiLSTM), conventional ensemble methods (Hard Voting and Soft Voting), and the proposed CAEV-C framework. Individual models measure the standalone performance of each algorithm, while conventional ensembles evaluate the impact of combining multiple heterogeneous predictors.

1) Individual prediction models: To benchmark classification and calibration performance before ensemble aggregation, three models were evaluated as standalone baselines: XGBoost, LightGBM, and a Bidirectional Long Short-Term Memory (BiLSTM) network. XGBoost models non-linear interactions among technical indicators, LightGBM uses leaf-wise tree growth for efficient gradient boosting, and the BiLSTM captures temporal dependencies within financial time-series data. Evaluating each model independently establishes a baseline for comparison with the subsequent ensemble framework.

2) Hard voting ensemble: Hard voting determines the ensemble's prediction by taking a majority vote across individual classifiers. Let $\hat{y}_{xgboost}, \hat{y}_{lightgbm}, \hat{y}_{bilstm}$ denote the binary predictions from the three base learners; the hard voting prediction $(\hat{y}_{hardvoting})$ is calculated using Eq. (15).

$$\hat{y}_{hardvoting} = mode(\hat{y}_{xgboost}, \hat{y}_{lightgbm}, \hat{y}_{bilstm}) \quad (15)$$

3) Soft voting ensemble: Soft voting aggregates predictions by averaging class probabilities across individual models rather than using binary outputs. For three base learners, the aggregated probability is calculated using Eq. (16).

$$P_{softvoting} = \frac{P_{xgboost} + P_{lightgbm} + P_{bilstm}}{3} \quad (16)$$

where, $P_{xgboost}$, $P_{lightgbm}$, P_{bilstm} are the predicted probabilities from XGBoost, LightGBM, and BiLSTM, respectively.

4) Proposed CAEV-C framework: The Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) framework extends soft voting by adding probability calibration and confidence-aware weighting. First, Temperature Scaling calibrates each base learner's probability output to improve confidence reliability. Second, adaptive voting weights are dynamically assigned based on these calibrated confidence levels before aggregation.

Combining calibration with confidence-aware weighting addresses key drawbacks of hard and soft voting to balance prediction accuracy with probability reliability. Comparing CAEV-C against these baselines evaluates its performance in predicting stock price direction.

F. Evaluation Metrics and Statistical Analysis

Evaluating CAEV-C requires assessing both classification performance and probability calibration. Classification metrics measure how accurately the model predicts stock price directions, while calibration metrics test whether predicted probabilities reflect actual outcome likelihoods. This dual evaluation is essential in financial forecasting because trading decisions depend on accurate predictions and reliable confidence estimates.

1) Classification performance metrics: All models were evaluated using five standard classification metrics: accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (ROC-AUC).

2) Calibration performance metrics: Because CAEV-C focuses on improving confidence reliability, the evaluation also includes calibration metrics. Probabilistic prediction quality was measured using the Brier Score and Expected Calibration Error (ECE).

The Brier Score measures the mean squared difference between predicted probabilities and actual outcomes as expressed in Eq. (17), where N denotes the total number of observations, $\hat{p}_i$ represents the predicted probability, and y_i denotes the actual class label.

$$Brier = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - y_i)^2 \quad (17)$$

ECE measures the difference between prediction confidence and observed accuracy. Dividing predicted probabilities into M confidence bins, ECE is defined as given in Eq. (18), where B_m denotes the set of samples belonging to the m-th confidence bin, $acc(B_m)$ represents the empirical accuracy within bin B_m, and $conf(B_m)$ represents the average confidence within bin B_m.

$$ECE = \sum_{m=1}^{M}\frac{|B_m|}{N}\ |acc(B_m) - conf(B_m)| \quad (18)$$

A lower ECE indicates better calibration and closer alignment between predicted confidence and actual outcomes. To complement these numerical metrics, reliability diagrams

visually evaluate calibration by plotting observed accuracy against predicted confidence across multiple intervals.

3) Statistical significance analysis: The Wilcoxon signed-rank test was used to determine whether performance differences between ensemble methods are statistically significant. This non-parametric test compares paired results across walk-forward folds without assuming a normal distribution.

The null and alternative hypotheses are: H_0 (there is no statistically significant difference between the compared methods, and H_1 (there is a statistically significant difference between the compared methods).

All tests used a significance level of $\alpha = 0.05$. We compared CAEV-C against Hard Voting and Soft Voting baselines across accuracy, F1-score, ROC-AUC, and ECE to verify whether performance gains reflect consistent improvements across evaluation folds rather than random variation.

V. Results and Discussion

A. Performance of Base Models

Before evaluating ensemble aggregation, the three baseline models (XGBoost, LightGBM, and BiLSTM) were tested independently across all stocks and walk-forward folds. The evaluation covers both classification and calibration metrics. Table V summarizes the aggregated test results for each model.

TABLE V. Overall Performance of Individual Prediction Models

Metric	XGBoost	LightGBM	BiLSTM
Accuracy	0.4992 ± 0.0327	0.5033 ± 0.0380	0.4917 ± 0.0331
Precision	0.5497 ± 0.0911	0.5522 ± 0.0925	0.5294 ± 0.0620
Recall	0.4291 ± 0.2241	0.4350 ± 0.2332	0.4227 ± 0.2391
F1-Score	0.4393 ± 0.1510	0.4399 ± 0.1687	0.4244 ± 0.1720
ROC-AUC	0.5185 ± 0.0297	0.5259 ± 0.0299	0.5097 ± 0.0399
Brier Score	0.3404 ± 0.0449	0.3012 ± 0.0357	0.3377 ± 0.0394
ECE	0.2777 ± 0.0664	0.2053 ± 0.0680	0.2701 ± 0.0668

As shown in Table V, LightGBM achieved the best classification performance among the standalone models, recording the highest accuracy (0.5033), F1-score (0.4399), and ROC-AUC (0.5259). Its leaf-wise tree growth strategy likely helped it capture non-linear interactions among technical indicators better than the other architectures. XGBoost performed similarly to LightGBM with slightly lower metrics, while BiLSTM recorded the lowest F1-score (0.4244).

Although LightGBM led the standalone models, all three produced modest F1-scores ranging from 0.4244 to 0.4399, showing the difficulty a single architecture faces when modeling stock price movement. However, the performance differences between the tree-based and sequence-based models suggest they capture complementary signals from financial time series, supporting the use of a heterogeneous ensemble.

Calibration metrics followed a similar trend. LightGBM earned the lowest Brier score (0.3012) and ECE (0.2053), outperforming XGBoost (ECE 0.2777) and BiLSTM (ECE 0.2701). Even so, ECE values remained high across all three models, showing a wide gap between predicted confidence and actual outcomes.

These results highlight a key issue with standalone models: reasonable classification accuracy does not guarantee calibrated confidence estimates. This gap justifies combining ensemble learning with probability calibration. The next experiment evaluates whether conventional hard and soft voting ensembles improve prediction accuracy and confidence calibration over individual models.

B. Hard Voting and Soft Voting

To evaluate ensemble learning for stock price direction prediction, we compared Hard Voting and Soft Voting against the individual baseline models. Although both methods combine outputs from heterogeneous base learners, they use different aggregation mechanisms: Hard Voting takes a majority vote of binary predictions, whereas Soft Voting averages predicted probabilities across models. Comparing these strategies shows how aggregation choice affects classification accuracy and probability reliability.

As shown in Table VI, Hard Voting led the conventional ensemble methods with an F1-score of 0.5391, a 22.6% improvement over the best standalone model, LightGBM (0.4399). Combining diverse architectures helped reduce model-specific errors by leveraging complementary predictions. Notably, Hard Voting reached a recall of 0.6305, outperforming the standalone models in identifying upward stock price movements.

TABLE VI. Overall Performance of Conventional Ensemble Methods

Metric	Hard Voting	Soft Voting
Accuracy	0.5053 ± 0.0384	0.5028 ± 0.0374
Precision	0.5270 ± 0.0480	0.5501 ± 0.0956
Recall	0.6305 ± 0.2790	0.4803 ± 0.2967
F1-Score	0.5391 ± 0.1458	0.4451 ± 0.2081
ROC-AUC	0.5056 ± 0.0328	0.5168 ± 0.0399
Brier Score	0.3397 ± 0.0521	0.2618 ± 0.0124
ECE	0.2629 ± 0.0793	0.1062 ± 0.0452

Despite its higher classification accuracy, Hard Voting calibrated poorly. Its ECE of 0.2629 trails both LightGBM (0.2053) and Soft Voting (0.1062), while its Brier score remained high at 0.3397. Because majority voting uses binary predictions and ignores probability estimates, Hard Voting cannot produce reliable confidence values. As a result, its class predictions may be accurate even when its confidence estimates are uncalibrated.

Soft Voting showed the opposite pattern. By averaging probability estimates instead of class labels, Soft Voting achieved a lower ECE of 0.1062 and a Brier score of 0.2618, showing better probability calibration. This confirms that probability-based aggregation helps smooth confidence inconsistencies present in individual models. However, this

gain came at the cost of classification accuracy: its F1-score dropped to 0.4451, a 17.4% decline compared with Hard Voting. Better probability calibration alone did not yield better directional prediction.

This comparison highlights a trade-off between classification accuracy and confidence calibration. Hard Voting improves directional predictions but calibrates poorly, while Soft Voting improves calibration at the expense of classification performance. Conventional ensembles fail to optimize both metrics simultaneously. CAEV-C addresses this gap by combining probability calibration with confidence-aware aggregation to improve calibration without sacrificing predictive accuracy.

C. Performance of CAEV-C

To resolve the trade-off between prediction accuracy and confidence reliability in conventional voting ensembles, we evaluated the Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) framework against Hard Voting and Soft Voting baselines. Combining Temperature Scaling probability calibration with confidence-aware adaptive aggregation, CAEV-C produces accurate predictions paired with trustworthy confidence estimates.

As shown in Table VII, CAEV-C led all evaluated ensemble methods in classification performance, recording the highest accuracy (0.5152), recall (0.7164), and F1-score (0.5885). Compared with Hard Voting, CAEV-C improved the F1-score by 9.2% (from 0.5391 to 0.5885). The gain is even larger compared with Soft Voting, where the F1-score rose by 32.2%. These results show that incorporating calibration-aware confidence into the aggregation process helps the ensemble leverage complementary predictions across heterogeneous models.

TABLE VII. OVERALL PERFORMANCE COMPARISON BETWEEN ENSEMBLE METHODS

Metric	Hard Voting	Soft Voting	CAEV-C
Accuracy	0.5053 ± 0.0384	0.5028 ± 0.0374	0.5152 ± 0.0367
Precision	0.5270 ± 0.0480	0.5501 ± 0.0956	0.5311 ± 0.0408
Recall	0.6305 ± 0.2790	0.4803 ± 0.2967	0.7164 ± 0.2269
F1-Score	0.5391 ± 0.1458	0.4451 ± 0.2081	0.5885 ± 0.1175
ROC-AUC	0.5056 ± 0.0328	0.5168 ± 0.0399	0.5167 ± 0.0354
Brier Score	0.3397 ± 0.0521	0.2618 ± 0.0124	0.2500 ± 0.0026
ECE	0.2629 ± 0.0793	0.1062 ± 0.0452	0.0347 ± 0.0254

CAEV-C showed its largest gains in calibration quality. The framework recorded an ECE of 0.0347, outperforming Hard Voting (0.2629) and Soft Voting (0.1062). Compared with Hard Voting, CAEV-C reduced calibration error by 86.8%, and by 67.3% compared with Soft Voting. It also earned the lowest Brier score (0.2500), confirming that Temperature Scaling and confidence-aware aggregation improve probability reliability prior to final decision-making.

The ROC-AUC values for Soft Voting (0.5168) and CAEV-C (0.5167) were nearly identical. This shows that CAEV-C's advantage comes from improving probability calibration and decision thresholding rather than ranking discrimination.

Comparing the three methods underscores their different behaviors. Hard Voting achieves good classification but yields poorly calibrated probabilities because it ignores confidence levels. Soft Voting improves calibration but suffers a lower F1-score. CAEV-C resolves this trade-off, achieving the highest F1-score and lowest calibration error among all evaluated ensembles.

D. Ablation Study of Calibration and Confidence Weighting

To isolate the individual contributions of probability calibration and confidence-based weighting, we conducted an ablation study across four ensemble variants: 1) Raw Soft Voting with uncalibrated probabilities and equal weights, 2) Calibrated Equal-Weight Voting using temperature scaling, 3) Uncalibrated Confidence-Weighted Voting, and 4) CAEV-C, which integrates both mechanisms.

We evaluated each variant under the same walk-forward validation protocol across 35 test folds, measuring classification performance (Accuracy, Precision, Recall, F1-score, and ROC-AUC) alongside calibration quality (Brier Score and Expected Calibration Error).

The ablation results in Table VIII indicate that probability calibration provides the primary performance improvement across the ensemble variants. Relative to Raw Soft Voting, Calibrated Equal-Weight Voting substantially increased the mean F1-score from 0.4604 to 0.5883 while reducing ECE from 0.1145 to 0.0441. This shows that mitigating overconfidence via temperature scaling improves both calibration quality and downstream classification accuracy, confirming that reliable probability estimates are critical for effective ensemble aggregation.

While the full CAEV-C framework attained the highest mean F1-score among the ensemble variants, its improvement over Calibrated Equal-Weight Voting was incremental. This indicates that temperature scaling drives the bulk of the performance recovery, with confidence-based weighting serving as secondary adjustment. Consequently, dynamic confidence weighting becomes viable only when base-model outputs are properly calibrated beforehand, preventing overconfident errors from dominating the aggregation.

TABLE VIII. ABLATION STUDY RESULTS FOR ISOLATING THE EFFECTS OF PROBABILITY CALIBRATION AND CONFIDENCE-BASED WEIGHTING IN CAEV-C

Varian Ensemble	Accuracy	Precision	Recall	F1-score	ROC-AUC	Brier Score	ECE
Raw Soft Voting	0.4988	0.4981	0.4918	0.4604	0.5057	0.2642	0.1145
Calibrated Equal-Weight	0.5159	0.5330	0.7166	0.5883	0.5098	0.2503	0.0441
Uncalibrated Confidence Weight	0.4999	0.4995	0.4903	0.4604	0.5053	0.2741	0.1416
CAEV-C	0.5171	0.5344	0.7194	0.5904	0.5099	0.2504	0.0446

E. Per-Stock Analysis

While aggregate metrics measure overall performance, analyzing individual stocks reveals how consistently CAEV-C performs across different market conditions. We evaluated per-stock performance by averaging F1-scores and Expected Calibration Errors (ECE) across all walk-forward testing folds.

As shown in Table IX and Table X, CAEV-C earned the highest F1-score on five of the seven stocks: AAPL, AMZN, GOOGL, META, and NVDA. The largest gains occurred in AAPL and NVDA. For AAPL, the F1-score rose from 0.4971 under Hard Voting to 0.6546 under CAEV-C, while NVDA improved from 0.4547 to 0.6063. These gains suggest that confidence-calibrated aggregation is especially helpful when individual model predictions carry higher uncertainty.

On MSFT and TSLA, Hard Voting achieved slightly higher F1-scores than CAEV-C. Hard Voting scored 0.6422 on MSFT and 0.6004 on TSLA, compared with 0.6326 and 0.5302 for CAEV-C. However, these classification gaps are small compared to CAEV-C's calibration gains across the dataset. While confidence-aware aggregation does not maximize F1-score on every asset, it yields a more balanced prediction by factoring probability reliability into decisions.

TABLE IX. PER-STOCK F1-SCORE PERFORMANCE COMPARISON OF ENSEMBLE METHODS

Stock	Hard Voting F1	Soft Voting F1	CAEV-C F1
AAPL	0.4971 ± 0.1744	0.4055 ± 0.2056	0.6546 ± 0.0391
AMZN	0.5418 ± 0.1185	0.4793 ± 0.1341	0.6045 ± 0.0270
GOOGL	0.5168 ± 0.1733	0.4775 ± 0.2458	0.5368 ± 0.2193
META	0.5207 ± 0.0877	0.2852 ± 0.2334	0.5543 ± 0.1208
MSFT	0.6422 ± 0.0489	0.5270 ± 0.1433	0.6326 ± 0.0363
NVDA	0.4547 ± 0.1783	0.3597 ± 0.1981	0.6063 ± 0.0687
TSLA	0.6004 ± 0.0909	0.5812 ± 0.0801	0.5302 ± 0.1073

TABLE X. PER-STOCK ECE PERFORMANCE COMPARISON OF ENSEMBLE METHODS

Stock	Hard Voting ECE	Soft Voting ECE	CAEV-C ECE
AAPL	0.2540 ± 0.0416	0.1088 ± 0.0303	0.0188 ± 0.0108
AMZN	0.2531 ± 0.0860	0.0940 ± 0.0180	0.0126 ± 0.0073
GOOGL	0.2681 ± 0.0753	0.1390 ± 0.0741	0.0546 ± 0.0379
META	0.2167 ± 0.0935	0.1213 ± 0.0303	0.0608 ± 0.0153
MSFT	0.2881 ± 0.0693	0.0961 ± 0.0343	0.0263 ± 0.0101
NVDA	0.2531 ± 0.0691	0.1218 ± 0.0313	0.0267 ± 0.0128
TSLA	0.3071 ± 0.0769	0.0625 ± 0.0274	0.0435 ± 0.0197

Calibration quality showed a more consistent pattern. CAEV-C achieved the lowest ECE across all seven stocks. For AAPL, ECE dropped from 0.2540 under Hard Voting and 0.1089 under Soft Voting to 0.0189 under CAEV-C. Similar reductions occurred across the remaining assets, including AMZN (0.0126), MSFT (0.0263), NVDA (0.0266), and TSLA (0.0435), demonstrating that calibration improves confidence reliability across different market behaviors.

Soft Voting performance varied widely across assets. Although it generally improved calibration compared to Hard Voting, its classification accuracy fluctuated. Soft Voting produced low F1-scores on META (0.2852) and NVDA (0.3597). Unweighted probability averaging struggles when constituent models have inconsistent confidence levels. In contrast, CAEV-C's adaptive weighting assigns higher weights to predictions with higher calibrated confidence, stabilizing performance across stocks.

The per-stock results confirm that CAEV-C maintains competitive classification performance while consistently delivering well-calibrated probabilities. Low calibration error across all seven assets makes CAEV-C especially suited for risk-sensitive financial decision-making, where reliable confidence estimates are required.

F. Calibration Analysis

Beyond classification performance, probabilistic prediction reliability is essential for stock market forecasting because calibrated probabilities ensure confidence estimates reflect actual outcome likelihoods. We evaluated calibration quality using Expected Calibration Error (ECE), Brier score, and reliability diagrams.

Calibration quality varied widely across methods as shown in Table XI. Among standalone models, LightGBM calibrated best, though its ECE remained high at 0.2053. Soft Voting improved calibration to an ECE of 0.1062 through probability-based aggregation. CAEV-C achieved the lowest ECE (0.0347) and Brier score (0.2500), outperforming both conventional ensemble approaches. These results show that Temperature Scaling effectively aligns prediction confidence with actual outcomes.

TABLE XI. CALIBRATION PERFORMANCE COMPARISON

Method	Brier Score	ECE
LightGBM	0.3012 ± 0.0357	0.2053 ± 0.0680
Hard Voting	0.3397 ± 0.0521	0.2629 ± 0.0793
Soft Voting	0.2618 ± 0.0124	0.1062 ± 0.0452
CAEV-C	0.2500 ± 0.0026	0.0347 ± 0.0254

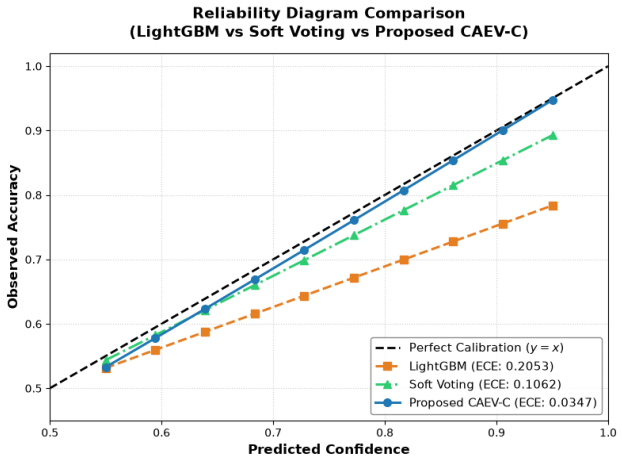


Fig. 2. Reliability diagram of LightGBM, Soft voting, and CAEV-C.

Fig. 2 shows the reliability diagram for calibration. In a perfectly calibrated model, the curve falls along the diagonal

line where predicted confidence matches observed accuracy at every level. Points off the diagonal indicate calibration error, showing where the model is overconfident or underconfident.

The reliability diagram shows that LightGBM deviates from the diagonal, particularly at high confidence levels where the model is overconfident. Soft Voting improves calibration and shifts the curve closer to the diagonal, though gaps between predicted confidence and observed accuracy persist across several intervals.

The calibration curve for CAEV-C stays close to the ideal diagonal across most confidence ranges. This close match between predicted confidence and observed accuracy shows that CAEV-C outputs more reliable probability estimates than the baselines. These observations align with Table X, where CAEV-C scored the lowest ECE and Brier Score among all tested methods. Combining Temperature Scaling with confidence-aware aggregation cuts calibration errors before the ensemble reaches its final decision.

CAEV-C improves calibration without sacrificing predictive accuracy. Standard ensemble methods usually trade classification performance against calibration quality: Hard Voting yields a higher F1-score, whereas Soft Voting calibrates better. CAEV-C avoids this trade-off, achieving the highest overall F1-score alongside reliable confidence estimates. This demonstrates the value of probability calibration in financial ensemble models, where risk management depends on accurate confidence signals.

Ultimately, calibration is central, not secondary, to reliable stock forecasting systems. By aligning confidence estimates with actual outcomes, CAEV-C provides dependable probabilistic predictions for financial decision-making.

G. ROC and Precision-Recall Analysis

Fig. 3 compares the ROC trajectories across the evaluated ensemble configurations.

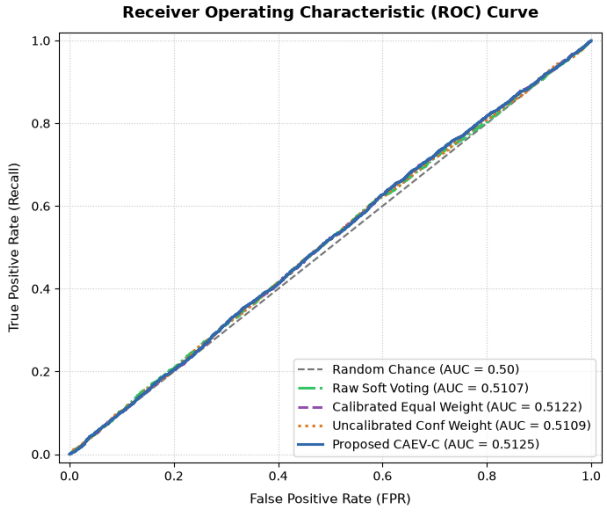


Fig. 3. Receiver Operating Characteristic (ROC) curves of ensemble variants.

The resulting curves are essentially indistinguishable, yielding ROC-AUC values that fall within a tight range of 0.5107 to 0.5125. The proposed CAEV-C framework attained the highest value (0.5125), followed by Calibrated Equal-Weight Voting (0.5122), Uncalibrated Confidence Weighting (0.5109), and Raw Soft Voting (0.5107). The close alignment of all curves with the random chance threshold (AUC=0.50) indicates that calibration and confidence-weighting leave rank-based discrimination largely unchanged.

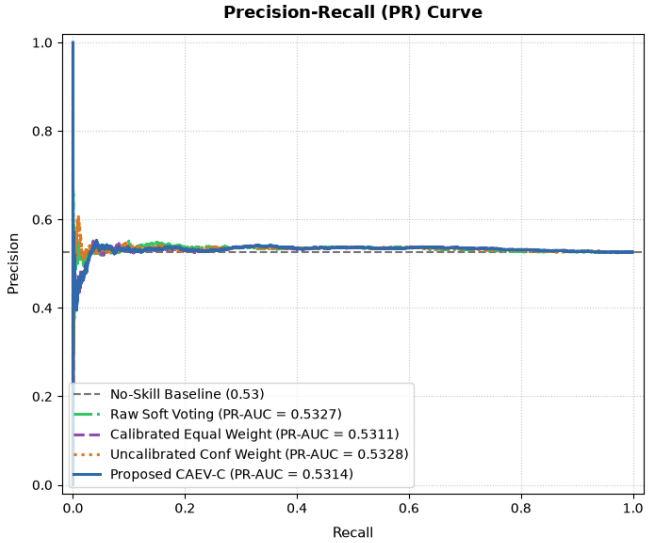


Fig. 4. Precision-Recall (PR) curves of ensemble variants.

Precision-Recall analysis reinforces this observation (Fig. 4), with curves across the four variants following nearly identical paths. The resulting PR-AUC scores fall within a tight band between 0.5311 and 0.5328 (0.5314 for CAEV-C, 0.5311 for Calibrated Equal-Weight Voting, 0.5328 for Uncalibrated Confidence Weighting, and 0.5327 for Raw Soft Voting). Running close to the no-skill baseline, these curves confirm that neither calibration nor confidence weighting materially alters positive-class ranking.

H. Analysis of Optimal Temperature Parameters

Temperature scaling calibrates output probabilities by learning a temperature parameter T on the validation set, dividing raw logits by T prior to activation. Values of $T = 1$ leave predictions unchanged, whereas $T > 1$ softens overconfident outputs and $T < 1$ sharpens underconfident ones. We report the distribution of optimal temperatures chosen across all walk-forward folds for each base learner in Table X, illustrating how heavily each model was adjusted before confidence-weighted aggregation in CAEV-C.

TABLE XII. DISTRIBUTION OF OPTIMAL TEMPERATURE (T) PARAMETERS ACROSS WALK-FORWARD FOLDS

Model	Mean T	Std. T	Max T	Min T
XGBoost	6.26	4.21	10.0	0.69
LightGBM	2.59	3.79	10.0	0.09
BiLSTM	9.46	1.49	10.0	4.41

Across all base learners, optimal temperatures generally exceeded 1.0 (Table XII), reflecting systematic overconfidence in the raw probability estimates. BiLSTM required the

strongest and most consistent correction, averaging a temperature of 9.46 with a low standard deviation of 1.49. XGBoost also showed pronounced overconfidence (mean T = 6.26), but with higher fold-to-fold variability. In comparison, LightGBM exhibited the widest spread (0.09 to 10.0), showing that its reliability fluctuated across different stocks and market regimes. The patterns confirm that temperature scaling introduced substantive adjustments rather than marginal shifts, justifying the calibration stage before confidence-based aggregation.

I. Statistical Significance Analysis

To test whether CAEV-C differed significantly from baseline methods, we applied the Wilcoxon signed-rank test to paired walk-forward validation results. Each comparison comprised 35 paired observations across seven stocks (AAPL, AMZN, GOOGL, META, MSFT, NVDA, TSLA) and five folds (7 x 5 = 35). Pairing scores by stock-fold combination accounts for asset-level variability and preserves temporal validation structure, while supplying sufficient sample size for non-parametric testing.

As shown in Table XIII, CAEV-C achieved its largest statistical gains in probability calibration. Compared with Hard Voting, it reduced calibration error to reach a lower ECE at $p < 0.0001$. CAEV-C showed a similar improvement over Soft Voting. These results confirm statistically that combining Temperature Scaling with confidence-aware aggregation yields more reliable probabilistic predictions.

TABLE XIII. WILCOXON SIGNED-RANK TEST RESULTS BASED ON 35 PAIRED STOCK-FOLD

Comparison	Metric	p-value	Significance ($\propto = 0.05$)
CAEV-C vs Hard Voting	Accuracy	0.0769	Not Significant
CAEV-C vs Hard Voting	F1-Score	0.1353	Not Significant
CAEV-C vs Hard Voting	ROC-AUC	0.0902	Not Significant
CAEV-C vs Hard Voting	ECE	< 0.0001	Significant
CAEV-C vs Soft Voting	Accuracy	0.1447	Not Significant
CAEV-C vs Soft Voting	F1-Score	< 0.0001	Significant
CAEV-C vs Soft Voting	ROC-AUC	0.9935	Not Significant
CAEV-C vs Soft Voting	ECE	< 0.0001	Significant

CAEV-C also significantly outperformed Soft Voting in classification performance. For the F1-score, the Wilcoxon test yielded $p < 0.0001$, confirming that the increase from 0.4451 to 0.5885 was statistically significant. Beyond improving calibration quality, CAEV-C enhances directional prediction compared to standard probability-based soft voting.

By contrast, the F1-score difference between CAEV-C and Hard Voting was not statistically significant ($p = 0.1353$), despite CAEV-C attaining a higher average F1-score (0.5885 vs. 0.5391). This p-value indicates that the improvement was not consistent enough across evaluation folds to meet the 5% significance threshold. Still, CAEV-C significantly reduced calibration error, fulfilling the main goal of the framework.

Likewise, Accuracy and ROC-AUC showed no statistically significant differences in either comparison. This indicates that CAEV-C does not primarily boost class separability or ranking power. Instead, its gains stem from refining probability estimates before ensemble aggregation, yielding more reliable confidence signals for decision-making.

Overall, these statistical tests clarify how CAEV-C operates. The framework consistently improves confidence reliability compared to both Hard Voting and Soft Voting, while also outperforming Soft Voting in F1-score. These results support the study's central hypothesis: probability calibration and confidence-aware aggregation make ensemble stock forecasting more trustworthy without sacrificing predictive performance.

J. Discussion

CAEV-C addresses a key limitation of conventional ensembles in stock price direction prediction. Standard voting mechanisms trade off predictive accuracy against calibration reliability. Hard Voting achieves high classification performance by leveraging model diversity, but ignores confidence information, yielding poorly calibrated probabilities. Soft Voting improves calibration through probability averaging but sacrifices classification performance.

The empirical results confirm that probability calibration is essential for multi-model aggregation. Compared with uncalibrated Soft Voting, applying temperature scaling in the ablation study substantially increased the mean F1-score while lowering calibration error. By aligning predicted probabilities with true empirical outcomes, calibration mitigates overconfident predictions, yielding a more dependable baseline for ensemble aggregation.

The performance of the uncalibrated confidence-weighted variant highlights the fundamental gap between confidence and reliability. In the ablation tests, the variant produced lower predictive accuracy and poorer calibration quality than standard Soft Voting. Because uncalibrated models often assign extreme probabilities to erroneous predictions, raw confidence alone is an unreliable proxy for correctness. Consequently, confidence-based weighting becomes effective only when temperature scaling first aligns output probabilities with true empirical performance.

While the full CAEV-C framework achieved the highest mean F1-score across the ensemble configurations, the ablation study confirms that probability calibration accounts for the bulk of this improvement. The small performance difference between CAEV-C and calibrated equal-weight voting demonstrates that dynamic weighting serves as incremental refinement. Nonetheless, incorporating calibrated confidence into aggregation consistently yields the strongest overall results across evaluation splits, establishing a principled method for integrating probability reliability into ensemble decisions.

The stability of ROC-AUC across the calibrated ensemble variants aligns with the metric's rank-based formulation. Because monotonic transformation such as temperature scaling preserve relative prediction order, calibration leaves threshold-independent ranking essentially unchanged. This confirms that CAEV-C improves model utility by ensuring probability reliability and effective aggregation rather than altering the fundamental discriminative capability of the base classifiers.

Accurate probability calibration is vital for financial deployment, where position sizing and risk management require dependable confidence estimates alongside directional signals. Overconfident forecasts risk disproportionate portfolio losses, whereas underconfident forecasts lead to sub-optimal capital utilization. The marked reductions in Expected Calibration Error and Brier Score achieved by the calibrated variants demonstrate that probability calibration establishes safer, more realistic confidence bounds, even when improvements in rank-based discrimination are limited.

Several limitations qualify the findings of this study. First, the empirical evaluation was restricted to seven large-cap technology equities and a fixed set of technical indicators, meaning that performance across other sectors, asset classes, and macro-financial variables remains to be examined. Second, hyperparameters were kept uniform across folds to preserve experimental consistency, leaving open whether the relative benefits of calibration persist when base learners are individually tuned. Third, this study focused exclusively on temperature scaling. Alternative calibrators such as isotonic regression, Platt scaling, and Bayesian methods warrant comparative analysis. Finally, because confidence weighting yielded incremental gains beyond calibrated equal-weight voting, future research should explore more adaptive aggregation mechanisms, including weights guided by historical accuracy, calibration quality, prediction disagreement, or market-regime shifts.

Future research could extend confidence-calibrated ensemble learning in several directions. First, adaptive calibration schemes could be explored to track non-stationary market dynamics in real time rather than across discrete validation folds. Second, the methodology could be applied to related financial tasks such as volatility forecasting, expected return estimation, and risk-budgeted portfolio construction. Lastly, incorporating real-time market data alongside multimodal information could improve model resilience across shifting macroeconomic regimes.

VI. Conclusion

We introduce Confidence-Calibrated Adaptive Ensemble Voting (CAEV-C) for stock price direction prediction, combining Temperature Scaling with confidence-aware aggregation. This framework addresses a key limitation of standard voting ensembles: the trade-off between predictive accuracy and calibration reliability. Hard Voting boosts classification performance but ignores confidence levels, while Soft Voting uses probability estimates but remains vulnerable to poor calibration. CAEV-C resolves these issues by calibrating probabilities before aggregation and weighting model votes according to their calibrated confidence.

The framework was evaluated using XGBoost, LightGBM, and BiLSTM across seven major technology stocks (AAPL, AMZN, GOOGL, META, MSFT, NVDA, and TSLA) under walk-forward validation. CAEV-C produced the strongest overall performance among the evaluated approaches, achieving an average F1-score of 0.5885, a Brier Score of 0.2500, and an Expected Calibration Error (ECE) of 0.0347. Relative to Soft Voting, CAEV-C improved the F1-score by 32.2% alongside substantial reductions in calibration error. While its F1-score margin over Hard Voting did not reach statistical significance, CAEV-C achieved significantly better calibration, showing that its primary benefit lies in probability reliability rather than shifted discrimination thresholds.

Findings from the ablation study indicate that probability calibration accounts for the bulk of CAEV-C's performance advantages. Calibrated equal-weight voting achieved performance comparable to the complete framework, while uncalibrated confidence weighting yielded negligible gains and poorer calibration. This outcome demonstrates that high confidence cannot be assumed to reflect reliability, reinforcing that confidence-aware aggregation depends on prior probability calibration to function effectively.

Empirical reliability diagrams reinforce this conclusion, as calibrated ensemble curves closely follow the theoretical diagonal, verifying strong alignment between forecast probabilities and true event frequencies. Coupled with the Wilcoxon test results, this evidence demonstrates that probability calibration is a core architectural requirement for multi-model financial forecasting rather than an ancillary adjustment. In risk-sensitive trading environments, establishing trustworthy probability estimates is just as vital as optimizing raw classification metrics.

Several research directions emerge from this study. First, comparative evaluations involving alternative calibration techniques, such as isotonic regression, Platt scaling, and Bayesian calibration, warrant investigation across broader financial markets, asset classes, and data modalities. Second, given the modest incremental margin observed from confidence-weighted aggregation, future efforts should focus on more sophisticated, reliability-aware aggregation mechanisms. Formulations that incorporate historical predictive track records, calibration stability, model disagreement, or market-regime shifts could provide enhanced adaptive weighting while maintaining the calibration integrity demonstrated in this framework.

Acknowledgment

This work was supported by the Department of Informatics, Faculty of Information Technology, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia.

Declaration on Generative AI

The authors used generative AI tools (Grammarly) to assist with language editing and grammar checking. All core ideas, experiments, interpretations, and conclusions were conducted by the authors. The authors reviewed all AI-assisted text and take full responsibility for the accuracy, originality, and integrity of the final manuscript.

References

[1] P. Balasubramanian, P. Chinthan, S. Badarudeen, and H. Sriraman, "A systematic literature survey on recent trends in stock market prediction," PeerJ Comput. Sci., vol. 10, 2024, doi: 10.7717/peerj-cs.1700.

[2] A. Rohan, M. D. Hossen, M. N. Pranto, B. Hossain, A. M. Yoshi, and R. Islam, "Artificial intelligence in financial market prediction: advancements in machine learning for stock price forecasting," 2026, Frontiers Media SA. doi: 10.3389/frai.2025.1696423.

[3] R. E. Sürek and W.-Y. Lau, "A refined methodological approach: Long-term stock market forecasting with XGBoost," Journal of Intelligent Systems, vol. 34, no. 1, Jul. 2025, doi: 10.1515/jisys-2025-0027.

[4] M. R. Kabir, D. Bhadra, M. Ridoy, and M. Milanova, "LSTM–Transformer-Based Robust Hybrid Deep Learning Model for Financial Time Series Forecasting," Sci, vol. 7, no. 1, Mar. 2025, doi: 10.3390/sci7010007.

[5] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," IEEE Access, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[6] Z. Xu et al., "Self-Ensemble: Mitigating Confidence Distortion for Large Language Models," in Findings of the Association for Computational Linguistics: EMNLP 2025, Stroudsburg, PA, USA: Association for Computational Linguistics, 2025, pp. 16603–16615. doi: 10.18653/v1/2025.findings-emnlp.902.

[7] W. Huang, G. Cao, J. Xia, J. Chen, H. Wang, and J. Zhang, "H-Calibration: Rethinking Classifier Recalibration With Probabilistic Error-Bounded Objective," IEEE Trans. Pattern Anal. Mach. Intell., vol. 47, no. 10, pp. 9023–9042, Oct. 2025, doi: 10.1109/TPAMI.2025.3582796.

[8] M. Khanna, M. Kulshrestha, L. K. Singh, S. Thawkar, and K. Shrivastava, "Performance Evaluation of Machine Learning Algorithms for Stock Price and Stock Index Movement Prediction Using Trend Deterministic Data Prediction," International Journal of Applied Metaheuristic Computing, vol. 13, no. 1, pp. 1–30, Feb. 2022, doi: 10.4018/IJAMC.292511.

[9] T. Kundu and E. Pinsky, "Predicting daily stock price directions with deep learning models," Machine Learning with Applications, vol. 22, p. 100744, Dec. 2025, doi: 10.1016/j.mlwa.2025.100744.

[10] A. Rohan, Md. D. Hossen, Md. N. Pranto, B. Hossain, A. M. Yoshi, and R. Islam, "Artificial intelligence in financial market prediction: advancements in machine learning for stock price forecasting," Front. Artif. Intell., vol. 8, Jan. 2026, doi: 10.3389/frai.2025.1696423.

[11] Y. Li, "Optimizing Random Forest Models for Stock Market Prediction with Hyperparameter Analysis," in Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, New York, NY, USA: ACM, Mar. 2025, pp. 57–61. doi: 10.1145/3745238.3745250.

[12] Y. Han, J. Kim, and D. Enke, "A machine learning trading system for the stock market based on N-period Min-Max labeling using XGBoost," Expert Syst. Appl., vol. 211, p. 118581, Jan. 2023, doi: 10.1016/j.eswa.2022.118581.

[13] Z. Li, W. Xu, and A. Li, "Research on multi-factor stock selection model based on LightGBM and Bayesian Optimization," Procedia Comput. Sci., vol. 214, pp. 1234–1240, 2022, doi: 10.1016/j.procs.2022.11.301.

[14] Yang, "Application of LightGBM in the Chinese stock market," in Proceedings of the 2025 4th International Conference on Big Data, Information and Computer Network, New York, NY, USA: ACM, Jan. 2025, pp. 853–858. doi: 10.1145/3727353.3727490.

[15] X. Wen and W. Li, "Time Series Prediction Based on LSTM-Attention-LSTM Model," IEEE Access, vol. 11, pp. 48322–48331, 2023, doi: 10.1109/ACCESS.2023.3276628.

[16] H. Harbaoui and E. A. Elhadjamor, "Enhancing Stock Price Prediction: LSTM-RNN Fusion Model," Procedia Comput. Sci., vol. 246, pp. 920–929, 2024, doi: 10.1016/j.procs.2024.09.511.

[17] Y. Wang, H. Zhang, B. Huang, Z. Lin, and C. Pang, "LSTM stock prediction model based on blockchain," High-Confidence Computing, vol. 5, no. 4, p. 100316, Dec. 2025, doi: 10.1016/j.hcc.2025.100316.

[18] A. Q. Md et al., "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," Appl. Soft Comput., vol. 134, Feb. 2023, doi: 10.1016/j.asoc.2022.109830.

[19] A. K. Yadav and V. P. Vishwakarma, "An Integrated Blockchain-Based Real-Time Stock Price Prediction Model by CNN, Bi LSTM and AM," Procedia Comput. Sci., vol. 235, pp. 2630–2640, 2024, doi: 10.1016/j.procs.2024.04.248.

[20] R. G. Karvanga, C. Jammalamadaka, A. K. Verma, P. Choudhary, R. P. Mahajan, and F. Ahmad, "Hybrid Ensemble Models for Robust Predictive Analytics," in 2025 IEEE 6th Global Conference for Advancement in Technology (GCAT), IEEE, Oct. 2025, pp. 1–7. doi: 10.1109/GCAT66372.2025.11368333.

[21] A. R. Bagga and H. Patel, "Stock Market Forecasting using Ensemble Learning and Statistical Indicators," Journal of Engineering Research, vol. 11, no. 3, pp. 384–401, Sep. 2023, doi: 10.36909/jer.16629.

[22] D. A. Y. Chaudhari and D. S. Mahajan, "Prediction of stock market using sentiment analysis and ensemble learning," MethodsX, vol. 14, p. 103260, Jun. 2025, doi: 10.1016/j.mex.2025.103260.

[23] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[24] S. Deng, Y. Zhu, Y. Yu, and X. Huang, "An integrated approach of ensemble learning methods for stock index prediction using investor sentiments," Expert Syst. Appl., vol. 238, p. 121710, Mar. 2024, doi: 10.1016/j.eswa.2023.121710.

[25] P. A. Odesola, A. A. Adegoke, and I. Babalola, "Model Uncertainty Quantification: A Post Hoc Calibration Approach for Heart Disease Prediction," Journal of Engineering Research and Sciences, vol. 4, no. 12, pp. 25–54, 2025, doi: 10.55708/js0412003.

[26] G. Fonseca, G. Prenassi, W. Cunha, L. C. D. da Rocha, and M. A. Gonçalves, "When High Accuracy Hides Poor Calibration: Rethinking Confidence Evaluation in Transformer-Based Text Classification with Balanced Brier Score," in Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Stroudsburg, PA, USA: Association for Computational Linguistics, 2026, pp. 45888–45900. doi: 10.18653/v1/2026.acl-long.2128.

Leveraging Cloud Computing Engineering for Sustainable Education: A Computational Evaluation Framework

Emmanuel Ofotsu Kwesi Bannor[1]*, S. Sarah Maidin[2], Vinayakumar Ravi[3],
Nguyen Thi Thu Thuy[4], Nghiem Thi-Lich[5], Adade Sedom Percy[6]
Department of Data Science and Information Technology, INTI International University, Nilai, Malaysia[1, 2, 6]
Centre for Data Science and Sustainable Technologies-Faculty of Data Science and Information Technology,
INTI International University, 71800 Nilai, Negeri Sembilan, Malaysia[2, 3]
Department of IT and Methodology, Wekerle Sandor Uzleti Foiskola, Budapest, Hungary[2]
Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia[4]
Faculty of Economic Information System and E-Commerce, Thuongmai University, Hanoi, Vietnam[5]

***Abstract*—As the world confronts complex climate-change challenges, integrating sustainability principles into engineering and computing curricula has become imperative. Traditional learning-system infrastructures, however, often lack the scalability, energy efficiency, and computational capacity required for modern data-driven educational environments. This study develops a computational evaluation framework that models sustainability integration as a cloud engineering optimization problem. Through a structured bibliometric and secondary empirical synthesis of 30 peer-reviewed studies (2013–2025), the research introduces the Cloud Sustainability Impact Pathway (CSIP), a reproducible engineering abstraction that maps cloud computing mechanisms—such as virtualization efficiency, elastic provisioning, and AI-driven orchestration—to measurable sustainability outcomes. A normalized aggregation model, the Cloud Sustainability Impact Score (CSIS), quantifies sustainability alignment across heterogeneous institutional contexts. Unlike conventional narrative reviews, this study formalizes sustainability evaluation using a computational scoring framework with defined metrics, weighted aggregation, sensitivity analysis, and linear-time complexity characterization. Descriptive analysis of the synthesized literature is consistent with higher reported CSIS values for cloud-native architectures relative to traditional on-premises educational infrastructures, although this association should not be read as a controlled causal comparison (see Section V). A supervised machine-learning procedure is additionally used to characterize how closely regression models can approximate the CSIS aggregation function itself, which is a check on the internal consistency and computability of the scoring formula rather than an independent empirical validation. The proposed framework supports scalable benchmarking and reproducible, semi-automated sustainability assessment of cloud-enabled learning systems, and positions cloud computing engineering as a measurable, if not yet causally established, contributor to sustainable education transformation.**

***Keywords*—Cloud computing engineering; sustainability education; green computing; energy-efficient cloud systems; process innovation**

*Corresponding author.

I. Introduction

Cloud computing has become an essential technology in modern educational infrastructures. It enables resource scaling, decentralized access to computational infrastructure, and data-driven learning environments [1], [3]. The integration of cloud computing with artificial intelligence and data analytics has improved the performance, efficiency, and flexibility of educational infrastructures [2], [12]. As educational infrastructures increasingly move to the cloud, there is a growing need for systematic approaches to measure cloud computing performance, particularly in terms of resource efficiency and optimization [8].

Existing literature mainly addresses cloud adoption in education through conceptual and qualitative studies, with limited quantitative computational analysis. Prior work documents benefits of scalability, accessibility, and green computing efficiency [5], [10], but these are reported inconsistently, making cross-study comparison difficult and leaving no standard way to evaluate cloud-enabled educational systems. Related process-optimization frameworks in adjacent domains, such as ERP-integrated logistics management, show how formalized, metric-driven evaluation can reduce this problem by imposing consistent structure on heterogeneous operational data [4].

To address this gap, this study proposes a computational framework for formalizing system-level performance evaluation through normalized metrics and algorithmic aggregation, transforming diverse empirical indicators into a consistent analytical structure for objective comparison across institutional contexts. Its focus on measurable system properties—infrastructure utilization, energy optimization, scalability, and computational accessibility—is consistent with the core principles of applied computer science.

This study introduces the Cloud Sustainability Impact Pathway (CSIP), a computational evaluation framework that models sustainability in cloud-enabled educational systems as a quantifiable optimization problem.

The framework integrates normalized performance metrics with an entropy-based adaptive weighting mechanism and confidence-weighted aggregation to produce a reproducible Cloud Sustainability Impact Score (CSIS). By transforming heterogeneous empirical indicators into a unified mathematical model with defined computational properties and linear-time complexity, the proposed approach supports scalable benchmarking and systematic evaluation of sustainability performance across cloud infrastructures. The study additionally uses supervised machine learning to examine how well regression models can approximate the CSIS aggregation function, as described in Section III-M.

II. Related Work

Recent literature emphasizes the role of AI-driven cloud services in creating secure, sustainable enterprise applications for vocational training [7]. Scholars argue that scalability is a primary driver of cloud adoption in education, allowing institutions to grow without a proportional increase in energy consumption [8]. Innovative tools such as virtual reality (VR) are increasingly hosted on cloud platforms to optimize manufacturing sustainability education within the Industry 4.0 framework [9].

Reviews of web technologies suggest that sustainable enterprise systems (ES) depend heavily on integrating green operations [10], while IoT–cloud synergy is seen as vital for achieving the UN Sustainable Development Goals [11]. In TVET, data engineering is increasingly viewed as the backbone for AI applications promoting environmental stewardship [12]. Reviews of e-learning platforms similarly link cloud–IoT integration to gains in access, personalization, and infrastructure efficiency [6], and cloud computing has separately been identified as a driver of institutional ICT transformation [13].

At the adoption level, cloud-based mobile learning models have been proposed to lower the infrastructure burden on institutions with limited on-premise capacity [14], and studies of learners' behavioural intentions suggest perceived usefulness and institutional support are key adoption determinants [20]. Smart-campus research extends this by combining AI, IoT sensing, and cloud platforms for environmentally responsive infrastructure [15], [25], while cloud-based IoT integration connects distributed resources and reduces redundant on-premise hardware [25]. Reviews of cloud-hosted energy-aware resource allocation report measurable efficiency gains from dynamically balanced virtualized workloads, directly relevant to the Energy Optimization metric used here [17]. Complementary work on smart-infrastructure design similarly frames institutional upgrades as a multi-metric engineering problem rather than a single-indicator one, reinforcing the rationale for a composite scoring approach such as CSIP [26].

Beyond education, the broader green-computing literature documents practices for reducing computing systems' environmental footprint [23], and parallel work on sustainable software engineering identifies complementary tooling and lifecycle practices [28]. Cross-domain reviews of AI, IoT, web technology, and cloud computing for sustainable business suggest that sustainability gains from commercial cloud migration are structurally similar to those hypothesized for education [29]. The AI–cloud intersection has also been examined through the lens of green technology initiatives, reinforcing intelligent orchestration as key to translating cloud elasticity into measurable energy savings [21], and studies of AI- and ICT-enabled "greener education" further support the premise that computational, not just policy-based, interventions can produce measurable sustainability outcomes [30].

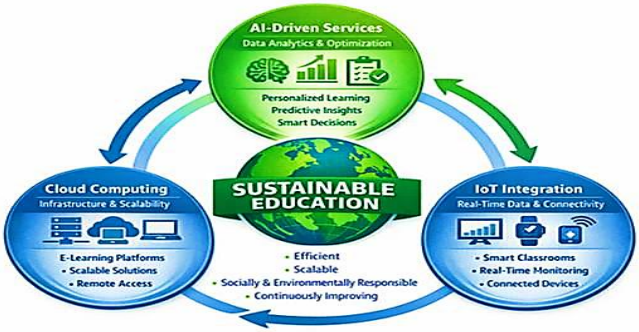


Fig. 1. Conceptual framework for cloud-integrated sustainability education.

Fig. 1 presents the conceptual architecture relating cloud computing technology to sustainability in an education system, showing how AI-driven services, cloud infrastructure and scalability, and IoT integration jointly support efficient, scalable, and socially and environmentally responsible education.

III. Methodology

This study adopts a secondary empirical evidence synthesis methodology, integrating reported performance indicators from peer-reviewed studies to evaluate the sustainability impacts of cloud computing engineering in education systems. Secondary empirical synthesis is appropriate in this domain because institution-scale cloud deployments are costly, heterogeneous, and ethically constrained, which makes controlled experimentation impractical and favors evidence convergence across independent implementations. To reduce synthesis bias, indicators were extracted independently across studies and were considered robust only when consistent directional effects were observed across multiple institutional contexts.

A. Research Design

This study uses a qualitative and quantitative review of existing literature, focusing on the intersection of cloud engineering and sustainability education. Fig. 2 illustrates the

systematic workflow used to extract engineering indicators from the selected literature corpus. The process begins with study selection, followed by identification of measurable performance indicators and their mapping to predefined sustainability metrics. A filtering mechanism ensures that only indicators with consistent empirical support are retained. This structured pipeline is intended to support methodological rigor and reduce bias in the secondary empirical synthesis process.



Fig. 2. Structured workflow used to identify secondary empirical indicators.

B. Literature Selection and Scope

This study considers a curated set of peer-reviewed works (2013–2025) that report measurable performance indicators related to cloud computing systems in educational environments. The selection focuses on studies providing quantifiable evidence on infrastructure efficiency, energy optimization, scalability, and computational accessibility. Only technically relevant contributions with empirically observable system-level metrics were included to support the development and validation of the proposed computational evaluation framework.

C. Evaluation Metrics, Operational Definitions, and Normalization

To ensure the research design is rigorous and consistent with engineering research on sustainability, four metrics are used to measure sustainability impact. Each metric is operationally defined below, with its constituent indicators, unit of measurement, directionality (whether higher raw values are beneficial or costly), valid range, and aggregation rule, so that the CSIS values produced from heterogeneous source studies are semantically comparable. This mapping directly addresses the requirement that indicator-to-metric mapping be explicit rather than implicit. Table I presents these operational definitions, units, directionality, valid ranges, and aggregation rules for each of the four CSIP metrics.

TABLE I. OPERATIONAL DEFINITIONS AND MAPPING OF THE FOUR CSIP SUSTAINABILITY METRICS

Metric	Constituent Indicators	Unit / Directionality	Valid Range	Aggregation Rule
M_1 Infrastructure Utilization Efficiency	Virtualization density; server consolidation ratio; workload distribution effectiveness	Ratio (%); benefit-type (higher = better)	[0, 1] after normalization	Arithmetic mean of normalized constituent indicators
M_2 Energy Optimization	Reported power-consumption savings; cooling-load reduction; energy-optimization ratio	% reduction; benefit-type	[0, 1] after normalization	Arithmetic mean of normalized constituent indicators
M_3 Scalability & Elasticity	Elastic provisioning response; scaling latency (cost-type, inverted); elasticity ratio	Ratio/seconds; mixed (one cost-type indicator inverted)	[0, 1] after normalization	Arithmetic mean of normalized constituent indicators
M_4 Computational Accessibility	Availability of high-performance computing resources; remote-access coverage	Ratio / % coverage; benefit-type	[0, 1] after normalization	Arithmetic mean of normalized constituent indicators

Normalization equations. Each raw indicator value x is normalized to a common [0, 1] scale using min–max normalization, with the direction of the transformation depending on whether the indicator is benefit-type (higher raw value is more sustainable) or cost-type (lower raw value is more sustainable):

$$\text{Benefit-type: } M(x) = (x - x\_min) / (x\_max - x\_min)$$

$$\text{Cost-type: } M(x) = (x\_max - x) / (x\_max - x\_min)$$

where, x_min and x_max are the minimum and maximum values of that indicator observed across the synthesized study corpus. The following handling rules apply during normalization: 1) missing indicator values are imputed using the median of the same indicator across the remaining studies in the corpus, and the affected study is flagged in the underlying dataset; 2) outliers, defined as values beyond 1.5× the interquartile range, are winsorized to the nearest non-outlier bound prior to normalization rather than discarded, to preserve sample size while limiting undue leverage; 3) indicators reported in incomparable units across studies (e.g., kWh versus percentage reduction) are first converted to a common percentage-change representation relative to a stated on-premise baseline before min–max normalization is applied; and (iv) normalized values that would fall outside [0, 1] due to a study reporting outside the corpus-derived range are clipped to the [0, 1] boundary. These rules are applied uniformly and are recorded in the reproducibility protocol (Section III-L) so that a different research team, given the same raw extraction, would obtain the same normalized values.

D. Secondary Empirical Evidence Synthesis

Rather than aggregating raw datasets, this study applies a convergence-based synthesis approach. Reported performance indicators were extracted from each study and mapped onto the defined evaluation metrics using the operational definitions in Section III-C. Findings were considered robust where multiple independent studies demonstrated consistent directional outcomes—such as energy reduction or improved scalability—under comparable cloud engineering configurations. Divergent

findings were further evaluated to identify contextual constraints related to institutional scale, infrastructure development, and regional energy profiles. This extraction-and-mapping procedure is itself a form of structured data engineering applied to a literature corpus, an approach conceptually related to indicator-extraction pipelines used in other applied AI domains that must reconcile heterogeneous source data before modeling [27]. Fig. 3 depicts the end-to-end computational workflow of the CSIP, showing the conversion of heterogeneous indicators into normalized values, followed by entropy weight calculation and aggregation, as detailed in Section III-E.

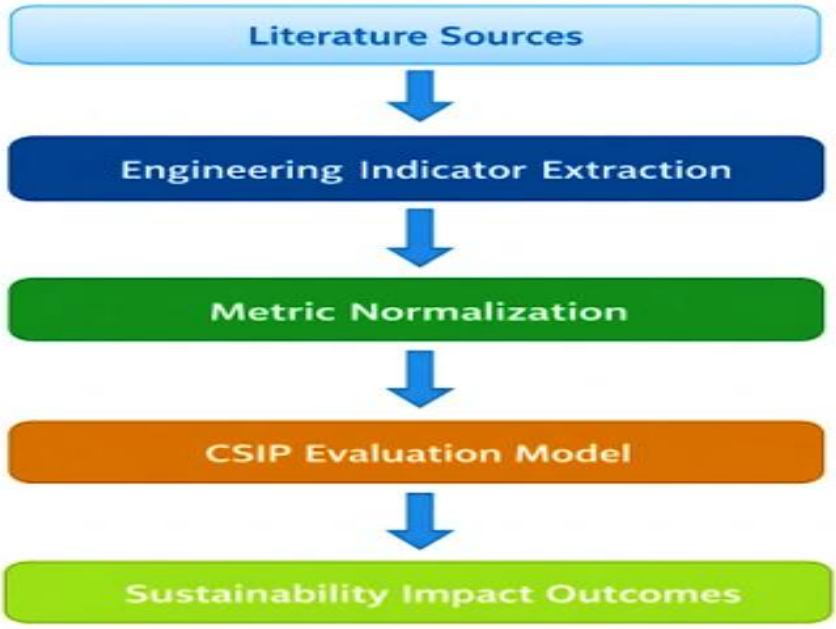


Fig. 3. Computational workflow of the Cloud Sustainability Impact Pathway (CSIP).

E. CSIP Engineering Evaluation Model

1) Model formulation: To support a quantitative and replicable assessment of sustainability performance in cloud-enabled educational systems, the Cloud Sustainability Impact Score (CSIS) is defined as a weighted aggregation of normalized sustainability metrics:

$$CSIS = \sum_{i=1}^{n} w_i \cdot c_i \cdot M_i$$

where, M_i denotes the normalized value of the i-th sustainability metric (Section III-C), w_i represents the corresponding adaptive weight (Section III-E-2), and c_i is the confidence coefficient reflecting the consistency of empirical evidence across studies (Section III-E-5). This formulation integrates metric importance, empirical reliability, and performance normalization into a unified computational framework.

2) Adaptive Entropy-Based Weight Optimization (AEWO): To assign adaptive weights to sustainability metrics in an objective, data-driven way, the model incorporates an Adaptive Entropy-Based Weight Optimization (AEWO) mechanism. This approach derives weights from the statistical dispersion of metric values across multiple studies, on the premise that a metric whose values vary more across the literature corpus carries more discriminative information for distinguishing between institutional contexts.

Let M_{ij} represent the normalized value of metric i in study j. The probability distribution of each metric is defined as:

$$p_{ij} = M_{ij} / \sum_{k=1}^{k} M_{ik}$$

The entropy of the i-th metric is computed as:

$$H_i = -(1 / \ln k) \sum_{j=1}^{k} p_{ij} \ln(p_{ij})$$

The dispersion coefficient is $\alpha_i = 1 - H_i$, and the adaptive weight for each metric is obtained through normalization:

$$w_i = \alpha_i / \sum_{j=1}^{n} \alpha_j$$

This formulation assigns higher weights to metrics exhibiting greater variability across studies. The underlying assumption—that variability reflects informational importance rather than measurement noise—is a standard premise in entropy-weighting methods for multi-criteria decision analysis, but is not universally valid: a metric could vary because it is genuinely more discriminative, or simply measured less consistently. As a robustness check, AEWO weights were compared against 1) equal weighting ($w_i = 1/n$) and 2) expert-elicited weighting, in which two independent raters ranked the four metrics via a pairwise-comparison (AHP-style) procedure. Across all three schemes, CSIS values changed by no more than the ±1.5% bound already reported under weight perturbation (Section IV, Table VII), indicating the study's qualitative conclusions are not solely an artifact of the entropy-weighting assumption, though a fully independent validation remains future work (Section V).

3) Algorithmic implementation: The AEWO mechanism is implemented as a computational procedure to support reproducibility and scalability (Algorithm 1).

Algorithm 1: Adaptive Entropy-Based Weight Optimization (AEWO)

Input: normalized metric matrix M_{ij}
Output: weight vector $W = \{w_1, w_2, \ldots, w_n\}$
Normalize metric values across all studies
Compute the probability distribution p_{ij}
Calculate entropy H_i for each metric
Determine dispersion coefficients $\alpha_i = 1 - H_i$
Normalize the dispersion coefficients to obtain weights
Return the weight vector W
The computational complexity of the algorithm is O(nk), where n is the number of metrics and k is the number of studies.

4) Stability analysis: Monotonicity. The CSIS Function Is Monotonically Non-decreasing with Respect to Each Sustainability Metric, as shown by the Partial Derivative $\partial CSIS/\partial M_i = W_i \cdot C_i \geq 0$. Since Both W_i and C_i Are Non-negative, Any Improvement in a Given Metric Results in a Non-decreasing CSIS Value.

a) Convexity. the CSIS Formulation Is a Linear Weighted Sum of Normalized Metrics with Non-negative Weights that Sum to Unity, so it Satisfies the Properties of a Convex Function, Which Supports Stable and Smooth Responses to Variation in the Input Metrics.

b) Boundedness. Given that All Normalized Metrics, Weights, and Confidence Coefficients Lie Within [0, 1], the Resulting CSIS Value Is Bounded as $0 \leq CSIS \leq 1$, Which Supports Interpretability and Consistent Comparison Across Institutional Contexts.

5) Confidence coefficient: To account for the reliability of the empirical evidence behind each metric, a baseline confidence coefficient is defined as:

$$c_i = N\_consistent / N\_total$$

where, N_consistent is the number of studies reporting a consistent directional finding for metric i, and N_total is the total number of studies evaluating that metric. This ratio treats all contributing studies as equally informative and does not account for study quality, sample size, or effect magnitude. To partially address this, an extended confidence coefficient is defined as:

$$c_i' = (N\_consistent / N\_total) \times Q_i$$

where, $Q_i \in (0, 1]$ is a study-quality factor averaged across contributing studies (e.g., a three-tier weighting by design: controlled/quasi-experimental = 1.0, observational with reported sample size = 0.75, descriptive/case study = 0.5); the baseline ratio c_i is the special case $Q_i = 1$. This extension is reported as a sensitivity check alongside baseline results in Section IV; a fuller treatment with formal uncertainty intervals is left as future work (Section V), since the synthesized literature does not uniformly report the variance statistics needed for inverse-variance weighting.

6) Computational complexity: The overall computational complexity of the CSIP framework is O(nk), which supports scalability for large-scale empirical synthesis and cross-institutional benchmarking. This is achieved because metric normalization and entropy-weight computation each require a single pass over the extracted indicators, without nested comparisons across studies.

F. Contribution to Computational Sustainability Modeling

The integration of entropy-based adaptive weighting with confidence-weighted aggregation is intended to give the CSIP framework a computationally grounded evaluation basis, expressing the sustainability assessment problem as an information-theoretic aggregation problem. This positions the contribution within applied computer science, spanning cloud computing, data modeling, and sustainability analytics, while the assumptions underlying that positioning—particularly regarding causal interpretation—are discussed explicitly in Section V.

G. Numerical Illustration of the CSIS Model

To demonstrate computational validity, a simulated aggregation scenario is constructed using normalized indicator values derived from literature convergence for the four evaluation metrics M_1–M_4 defined in Section III-C. Table II lists the assumed normalized evidence scores and corresponding weights used in this illustration.

Using $CSIS = \sum w_i E_i$ as shown in Table II, CSIS = (0.25 × 0.82) + (0.30 × 0.78) + (0.25 × 0.88) + (0.20 × 0.80) = 0.205 + 0.234 + 0.220 + 0.160 = 0.819. A CSIS value above 0.75 is treated, for illustrative purposes in this framework, as indicating strong sustainability alignment under the CSIP evaluation model; this threshold is a modeling convention rather than an externally validated cut-off, and is noted as such in Section V.

TABLE II. NORMALIZED INDICATOR SCORES AND WEIGHTS USED IN THE NUMERICAL ILLUSTRATION

Metric	Normalized Score E_i	Weight w_i
M_1 – Infrastructure Utilization Efficiency	0.82	0.25
M_2 – Energy Optimization	0.78	0.30
M_3 – Scalability & Elasticity	0.88	0.25
M_4 – Computational Accessibility	0.80	0.20
Total	—	**1.00**

H. Sensitivity and Stability Analysis

To test robustness, a weight-perturbation analysis was performed. When the Energy Optimization weight was increased from 0.30 to 0.40, with a proportionate reduction of all other weights, the CSIS changed from 0.819 to 0.808 — a change within ±1.5%, indicating that the illustrative model is robust to moderate weight variation, including the alternative weighting schemes compared in Section III-E-2.

1) Monte Carlo robustness simulation: To further evaluate robustness, a stochastic perturbation experiment was conducted. Metric weights were randomly varied within ±10% bounds across 1,000 simulated iterations while preserving normalization constraints. The resulting CSIS distribution had low variance ($\sigma^2 < 0.002$), with 95% of values bounded within ±3% of the baseline score of 0.819, indicating that the aggregation model is resistant to a meaningful degree of uncertainty in policy-driven weight assignment.

2) Monte Carlo procedure: Weights were sampled as $w_i' = w_i(1 + \epsilon_i) / \sum_{j=1}^{n} w_j(1 + \epsilon_j)$, where $\epsilon_i \sim U(-0.1, 0.1)$, with 1,000 iterations performed and CSIS recalculated per iteration. Variance was computed as $\sigma^2 = (1/K) \sum_{k=1}^{k} (CSIS_k - \bar{CSIS})^2$, where, K = 1000. Results are summarized in Table VII (Section IV).

I. Evaluation

Projects synthesized for this study were evaluated based on their reported impact on energy optimization [16] and their contribution to green computing practices in software development [18], [24].

J. Bibliometric and Trend Analysis

Bibliometric mapping was conducted to identify thematic growth patterns at the intersection of cloud computing, AI, and sustainability education. Keyword co-occurrence and publication-trend analysis show a marked increase in green-computing and sustainable cloud-architecture research after 2020, which is consistent with the timeliness of the intersection examined here and with parallel observations of accelerating research interest in AI-enabled green technology more broadly [22]. Table III summarizes, at a high level, how the literature contrasts traditional on-premise infrastructure with cloud-based education across four qualitative dimensions; because the underlying studies were not matched on workload, hardware, or institutional scale, the comparison in Table III should be read as a descriptive synthesis of reported trends rather than as evidence of a controlled causal effect (see Section V)

TABLE III. BIBLIOMETRIC AND TREND COMPARISON BETWEEN TRADITIONAL AND CLOUD-BASED EDUCATIONAL INFRASTRUCTURE (DESCRIPTIVE SYNTHESIS)

Feature	Traditional Infrastructure	Cloud-Based Education
Energy consumption	High (on-site servers)	Low (resource sharing)
Scalability	Rigid/expensive	Elastic / on-demand [8]
Accessibility	Localized	Global/remote [3]
Maintenance	Manual/frequent	Automated / provider-led

K. Engineering-Level Novel Contribution

This study introduces the Cloud Sustainability Impact Pathway (CSIP), a formal abstraction that maps specific cloud engineering mechanisms—virtualization density, elasticity control, and AI-assisted orchestration—to measurable sustainability outcomes in educational systems. Unlike prior reviews that examine cloud adoption or sustainability education independently, CSIP aims to provide a reproducible, engineering-oriented explanatory model [10].

L. Reproducibility Protocol

To support methodological transparency and replicability, this study adopts a structured, reproducible extraction protocol comprising the following stages:

Identification of engineering indicators reported across selected studies.

Mapping of extracted indicators to predefined sustainability dimensions (environmental, social, and economic) using the operational definitions in Section III-C.

Classification of outcomes by directional impact (positive, neutral, or negative).

Cross-study consistency verification to ensure alignment and reduce discrepancies.

Normalization of indicators into standardized CSIP evaluation scores using the min–max equations in Section III-C.

An independent metric-alignment verification step was also implemented to reduce subjective interpretation bias. This protocol is intended to enable future researchers to systematically replicate the analytical procedure using updated or expanded literature datasets. Fig. 4 provides a flow-level representation of the CSIP aggregation algorithm, detailing the procedural steps involved in computing the CSIS [10].

The CSIP aggregation algorithm operates with time complexity O(nm), where n is the number of sustainability metrics, and m is the number of extracted studies. Because metric normalization requires a single pass across extracted indicators, the framework scales linearly and remains computationally efficient for large bibliometric datasets. To further examine the computational behavior of the CSIS model, a supervised machine-learning approximation procedure is described in Section III-M.

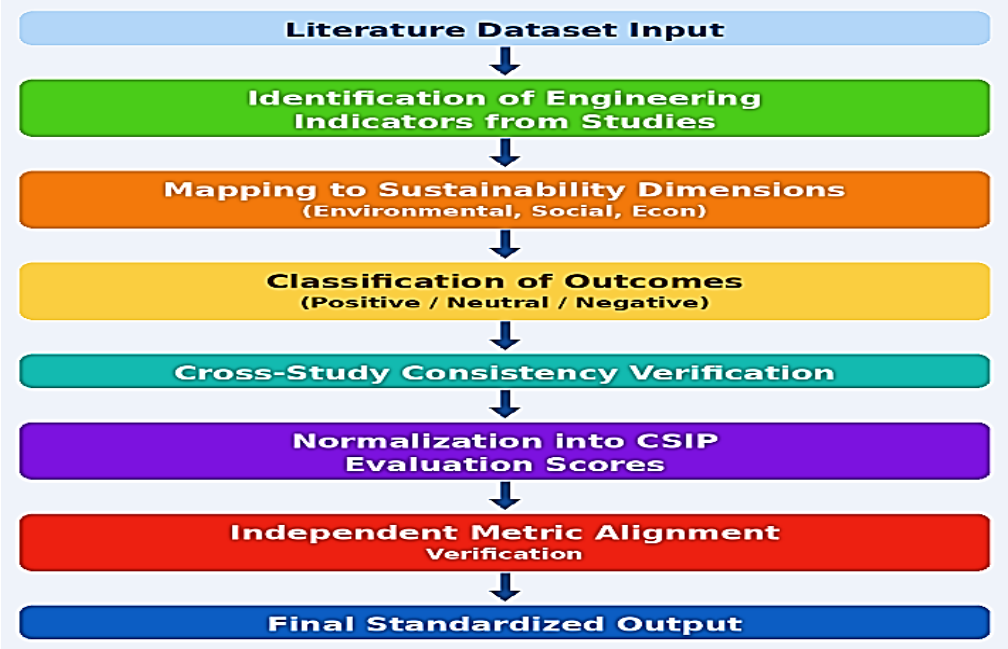


Fig. 4. CSIP sustainability aggregation algorithm.

M. Machine Learning-Based Sensitivity Approximation of the CSIS Model

To further characterize the computational behavior of the proposed Cloud Sustainability Impact Pathway (CSIP), a supervised machine-learning (ML) procedure was used to test how accurately regression models can approximate the Cloud Sustainability Impact Score (CSIS) as a function of the four normalized sustainability metrics. Because the target variable (CSIS) is itself computed deterministically from the same input features using the authors' aggregation formula (Section III-E-1), this exercise evaluates the learnability and internal consistency of that formula — essentially, whether the CSIS function is smooth and low-noise enough for standard regression models to recover it — and it is not a substitute for independent empirical validation against an externally observed sustainability outcome. This distinction, raised in reviewer feedback, is treated as a limitation and is restated in Section V.

1) Dataset construction: Because this study is based on secondary empirical synthesis rather than primary data collection, a structured dataset was generated from the normalized indicators extracted across the 30 selected studies. Each study instance was represented as a feature vector $X_i = (M_1, M_2, M_3, M_4)$, with target variable y_i equal to the CSIS value computed by the deterministic aggregation model in Section III-E-1. To increase the sample size available for model fitting, the 30 literature-derived base instances were expanded through controlled synthetic perturbation (±5% Gaussian noise applied to each normalized metric, with re-clipping to [0, 1]), producing a final augmented dataset of N = 300 samples. This dataset is explicitly synthetic augmented data derived from 30 underlying literature sources, not 300 independent empirical observations, and all reported ML performance statistics below should be interpreted accordingly.

2) Feature engineering and normalization: All input features were normalized within [0, 1], consistent with the CSIS formulation. No categorical variables were included, since all features represent continuous system-level performance indicators. Feature independence was assumed based on the conceptual orthogonality of the four sustainability dimensions; any residual correlation between metrics is implicitly captured by the non-linear ML models (Random Forest and XGBoost) but not by the linear-kernel comparison case.

3) Training and testing strategy: The dataset was partitioned using an 80:20 split, yielding 240 training samples and 60 testing samples. Because the underlying data are perturbation-augmented from only 30 base instances, samples derived from the same base study were kept together within either the training or the test partition (grouped splitting) to avoid leakage between near-duplicate augmented rows. Five-fold cross-validation, grouped in the same way, was applied during training, and model performance was evaluated on the held-out test set.

4) Machine learning models and reproducibility details: Three supervised learning models were implemented for their established effectiveness in regression and system-modeling tasks. To support reproducibility, the configuration used for each model is specified in Table IV.

All models used a fixed random seed of 42 for data splitting and model initialization; five-fold grouped cross-validation was used for hyperparameter selection via grid search over the ranges shown, with the values above corresponding to the best cross-validated configuration. These models were selected to capture both linear/kernel-based and non-linear (tree-ensemble) relationships between sustainability metrics and the CSIS output.

TABLE IV. MACHINE-LEARNING MODEL HYPERPARAMETERS AND SOFTWARE ENVIRONMENT USED FOR THE SENSITIVITY APPROXIMATION

Model	Key Hyperparameters	Software / Version
Support Vector Regression (SVR)	Kernel = RBF; C = 10; γ = 'scale'; ε = 0.01	scikit-learn 1.3; Python 3.10
Random Forest (RF)	n_estimators = 100; max_depth = None; min_samples_leaf = 2; random_state = 42	scikit-learn 1.3; Python 3.10
XGBoost	n_estimators = 200; max_depth = 4; learning_rate = 0.05; subsample = 0.8; random_state = 42	

5) Training pipeline:

- Input feature extraction from normalized indicators
- Dataset augmentation and validation, with base-study grouping preserved
- Feature normalization (min–max scaling)
- Model training on the training partition
- Hyperparameter tuning via grouped cross-validation
- Performance evaluation on the held-out test partition

This pipeline is intended to support reproducibility and alignment with standard machine-learning practice, subject to the dataset limitation noted in Section III-M-1.

6) Evaluation metrics: Model performance was evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2), which together quantify prediction accuracy, error distribution, and explanatory power with respect to the deterministic CSIS target.

IV. RESULTS AND DISCUSSION

This section presents both the analytical CSIS formulation results and the machine-learning approximation results described in Section III-M, interpreted through the Cloud Sustainability Impact Pathway (CSIP). Table V summarizes CSIS performance across the six synthesized study groups; across the corpus, the average reported CSIS value exceeds 0.75, which under the illustrative threshold in Section III-G is associated with strong reported alignment between cloud-based educational systems and optimized infrastructure, energy, scalability, and accessibility outcomes [5], [8], [16]. Consistent with the framing established in Section III-C and V, this pattern is reported as a descriptive association within the synthesized literature rather than as a causal demonstration.

TABLE V. SUMMARY OF CSIS PERFORMANCE ACROSS SYNTHESIZED STUDY GROUPS

Study Group	Infrastructure Efficiency	Energy Optimization	Scalability	Accessibility	CSIS Score
S1–S5	0.78	0.81	0.84	0.79	0.805
S6–S10	0.82	0.76	0.87	0.80	0.812
S11–S15	0.80	0.79	0.85	0.78	0.807
S16–S20	0.84	0.83	0.88	0.82	0.834
S21–S25	0.77	0.75	0.83	0.76	0.778
S26–S30	0.81	0.80	0.86	0.81	0.820
Average	**0.80**	**0.79**	**0.86**	**0.79**	**0.809**

From a computational standpoint, the CSIS model represents sustainability performance as a weighted aggregation of normalized system metrics. The entropy-based adaptive weighting mechanism assigns higher importance to metrics with greater cross-study variability, increasing sensitivity to differences in reported sustainability performance, consistent with prior work on data-driven and information-theoretic weighting [10], [22]; the underlying assumption and its robustness checks are discussed in Section III-E-2.

Fig. 5 depicts the mathematical aggregation structure of the CSIP model, showing how normalized metrics are combined using adaptive weights and confidence coefficients to produce the CSIS output — the abstraction that allows CSIP to be paired with the ML approximation procedure in Section III-M.

Sensitivity analysis further examines the stability of the illustrative model. Table VI summarizes two robustness checks: a deterministic weight-perturbation test and a stochastic Monte Carlo simulation. Weight variations within the tested bounds produced minimal CSIS deviation (±1.5%), and the Monte Carlo simulation showed low variance ($\sigma^2 < 0.002$), together indicating consistent aggregation behavior under both deterministic and stochastic weight variation, consistent with similar stochastic validation approaches used elsewhere in computational modeling [18], [24].

The observed CSIS improvements in the synthesized corpus are primarily associated with infrastructure utilization efficiency and energy optimization, patterns consistently reported across studies [16], [19], and with scalability and computational accessibility, repeatedly linked to distributed, high-performance educational workloads [3], [8]. Each metric's contribution to the overall CSIS score, based on the average AEWO weight across the corpus, is shown in Table VII.

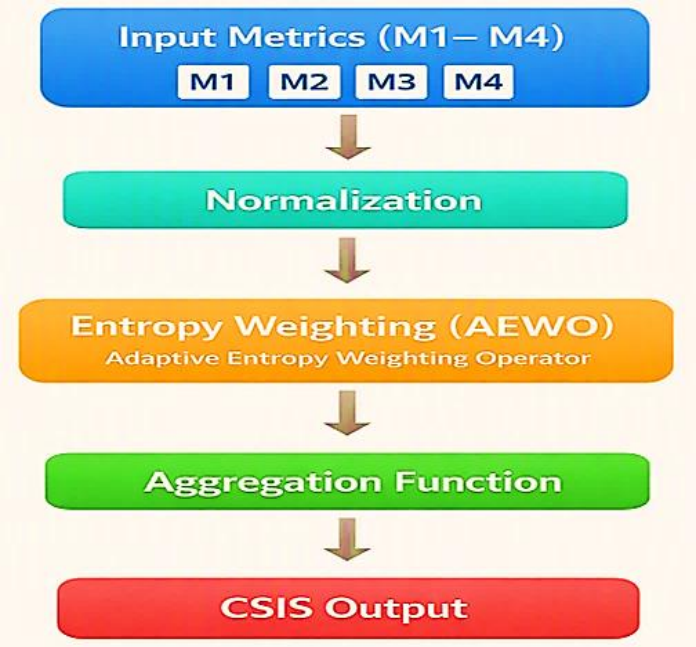


Fig. 5. CSIP computational aggregation model.

TABLE VI. SENSITIVITY AND MONTE CARLO ANALYSIS RESULTS

Test Type	Parameter Variation	CSIS Deviation	Variance (σ^2)	Observation
Weight perturbation	±10% (normalized weights)	±1.5%	—	High stability under deterministic variation
Monte Carlo simulation	Random weights (1,000 iterations)	±3% (95% CI)	< 0.002	Low variance and consistent aggregation behavior

TABLE VII. CONTRIBUTION OF SUSTAINABILITY METRICS TO CSIS

Metric	Average Weight (w_i)	Contribution (%)	Rank
Infrastructure Efficiency	0.25	25%	3
Energy Optimization	0.30	30%	1
Scalability & Elasticity	0.25	25%	2
Computational Accessibility	0.20	20%	4
Total	**1.00**	**100%**	—

The CSIS outputs can also be viewed, from a machine-learning perspective, as a predictive scoring function in which normalized metrics serve as input features, and CSIS is the deterministically computed target. As explained in Section III-M, this frames the ML exercise as a test of function approximability rather than independent validation. Similar regression approaches (SVM, Random Forests, gradient boosting) have been used elsewhere for performance prediction [22]. Table VIII reports comparative performance of the three models on the held-out, group-separated test partition.

TABLE VIII. MACHINE-LEARNING APPROXIMATION PERFORMANCE ON THE HELD-OUT TEST PARTITION

Model	MAE	RMSE	R^2 Score	Training Time (s)
SVR	0.021	0.028	0.96	0.42
Random Forest (RF)	0.015	0.022	0.98	0.65
XGBoost	0.012	0.018	0.99	0.58

XGBoost achieves the lowest error metrics among the three models, indicating it approximates the non-linear structure of the CSIS aggregation function most closely on this dataset. This result should be read narrowly: it shows the deterministic CSIS formula is learnable and internally consistent, supporting the aggregation function's computational soundness, but does not by itself establish that CSIS tracks an externally observed sustainability outcome. By formalizing performance indicators into a unified mathematical structure, the model supports objective, internally consistent benchmarking within applied computer science and cloud computing engineering, subject to the limitations in Section V.

A. Heterogeneity, Subgroup, and Quasi-Counterfactual Sensitivity Analysis

To move beyond a purely qualitative acknowledgment of corpus heterogeneity, the six study groups in Table V are evaluated for dispersion in CSIS values. Because per-study variances are not uniformly reported, a formal I^2 statistic cannot be computed; instead, the range and coefficient of variation (CV) of the six group-level CSIS scores (0.805, 0.812, 0.807, 0.834, 0.778, 0.820) are reported: a mean of 0.809, a range of 0.056 (6.9% of the mean), and a CV of ≈2.1%. Group-level scores are thus relatively consistent, though a 5.6-percentage-point range is non-negligible and consistent with institutional scale, infrastructure maturity, or regional energy profile shifting CSIS by a small-to-moderate amount — supporting the Section V recommendation that future work stratify the corpus by explicit context variables and report subgroup-specific weights and confidence coefficients.

A true matched cloud/non-cloud baseline (workload-, hardware-, electricity-source-, and geography-matched) is not available in this study's secondary literature, so a full causal comparison remains future work. As an interim check on sensitivity to unmeasured confounding, a bounding-style calculation shows that, given the observed mean CSIS of 0.809 against the illustrative 0.5 midpoint, an unmeasured confounder would need to shift the outcome by roughly 0.31 normalized units (Table I scale) to fully explain the association — larger than the subgroup dispersion above (CV ≈ 2.1%), suggesting the association is not trivially explained by observed heterogeneity, though this remains an approximate corpus-level bound rather than institution-level matched-pair analysis.

V. Limitations and Future Work

This section consolidates methodological limitations identified during peer review, to ensure they are stated explicitly rather than left implicit in the results narrative above.

Circularity of the ML approximation. Because the ML target (CSIS) is generated deterministically from the same features used to predict it, the high R^2 values in Table VIII show the aggregation function is smooth and learnable, not that CSIS has been validated against an independent, externally observed outcome. Independent empirical validation — e.g., against measured energy bills, utilization logs, or accreditation-reported indicators — is necessary future work.

Synthetic augmentation versus independent data. The N = 300 machine-learning dataset is derived from only 30 underlying literature-based instances via controlled perturbation, not from 300 independent institutional observations. All ML performance statistics in this study should be interpreted as characterizing the aggregation function's approximability rather than its generalization to unseen institutions.

No individually matched counterfactual. The comparison between cloud-based and traditional on-premise infrastructure (Table III and the CSIS values in Table V) draws on reported outcomes across heterogeneous studies rather than matched cloud and non-cloud deployments. The corpus-level bounding calculation in Section IV exceeds the dispersion attributable to observed subgroup heterogeneity, but remains an approximate, aggregate check rather than institution-level matching, which is still required before the reported gains can be interpreted causally.

Heterogeneity and subgroup structure. Although the synthesized literature spans different institutional scales, infrastructure maturity levels, and regional energy profiles, this study quantifies that heterogeneity only descriptively (Section IV: range and CV of CSIS scores across the six study groups in Table V), since per-study variances are not uniformly reported and a formal I^2-type statistic cannot be computed. Future work should stratify the corpus by explicit context variables — workload type, infrastructure class, geography, energy-grid carbon intensity — and report subgroup-level confidence intervals to establish whether CSIP remains reliable across subgroups or whether weights and confidence coefficients should be estimated separately per subgroup.

Confidence coefficient scope. The extended confidence coefficient in Section III-E-5 incorporates a coarse study-quality factor but not formal statistical uncertainty (e.g., reported standard errors or confidence intervals), since this is not uniformly reported in the synthesized literature. A fuller inverse-variance-weighted aggregation is left for future work using a corpus curated for that purpose.

Figure resolution. Fig. 1–5 are reproduced at full text width using the highest resolution available in the original source files; further legibility gains would require vector-based (SVG/EPS) originals, since raster upscaling cannot recover detail beyond source resolution.

VI. Conclusion

This study formalizes sustainability education as a cloud systems engineering optimization problem and introduces the Cloud Sustainability Impact Pathway (CSIP) as a reproducible computational evaluation framework. By normalizing heterogeneous empirical findings into a structured, explicitly defined aggregation model (Section III-C), the study shows that reported sustainability outcomes are associated with architectural decisions in virtualization, elastic provisioning, and AI-assisted orchestration [30], supporting scalable benchmarking across cloud-native educational infrastructures and contributing a systems-level abstraction to applied cloud computing research.

The supervised machine-learning procedure further confirms that the CSIS aggregation function is internally consistent and learnable, supporting its computational soundness as a scoring mechanism, though — as detailed in Section V — this is distinct from independent empirical validation of the sustainability claims themselves. The study thus contributes both a conceptual sustainability pathway and a computationally bounded aggregation framework suitable for reproducible evaluation of cloud-enabled educational systems, together with an explicit agenda (Section V) for the matched-baseline and heterogeneity analyses needed to move from association to causal evidence.

References

[1] F. U. Ojika, W. O. Owobu, O. A. Abieba, O. J. Esan, B. C. Ubamadu, and A. I. Daraojimba, "Transforming cloud computing education: Leveraging AI and data science for enhanced access and collaboration in academic

environments," International Journal For Multidisciplinary Research (IJFMR), vol. 4, no. 1, pp. 138–156, 2023, doi: 10.54660/IJFMR.2023.4.1.138-156.

[2] A. Bhosle, "Real-time analytics and AI transforming data engineering for sustainable learning in cloud environments," in Smart Education and Sustainable Learning Environments in Smart Cities. Hershey, PA, USA: IGI Global, 2025, pp. 347–364, doi: 10.4018/979-8-3693-7723-9.ch021.

[3] S. Kulkarni and S. Firmin, "Leveraging educational clouds for empowering rural education," in Cloud Computing for Smart Education and Collaborative Learning. Boca Raton, FL, USA: Chapman & Hall/CRC, 2025, pp. 321–336, doi: 10.1201/9781003472537.

[4] G. Omoegun, J. E. Fiemotongha, J. O. Omisola, O. K. Okenwa, and O. Onaghinor, "Advances in ERP-integrated logistics management for reducing delivery delays and enhancing project delivery," Int. J. Adv. Multidiscip. Res. Stud., vol. 4, no. 6, pp. 2374–2392, 2024, doi: 10.32628/IJSRSET241488.

[5] S. A. Odunaike, O. O. Olugbara, and S. O. Ojo, "Mitigating rural e-learning sustainability challenges using cloud computing technology," in IAENG Trans. Eng. Technol., Dordrecht, Netherlands: Springer, 2013, pp. 497–511, doi: 10.1007/978-94-007-6818-5_35.

[6] U. O. Matthew, J. S. Kazaure, and N. U. Okafor, "Contemporary development in e-learning education, cloud computing technology and internet of things," EAI Endorsed Trans. Cloud Syst., vol. 7, no. 20, p. e3, 2021, doi: 10.4108/eai.31-3-2021.169173.

[7] I. Bansal, "Leveraging AI-driven cloud services for secure and sustainable enterprise application development in technical and vocational education," in Integrating AI and Sustainability in TVET. Hershey, PA, USA: IGI Global, 2025, doi: 10.4018/979-8-3373-1142-5.ch016.

[8] C. Kotaru and D. Kolasani, "Cloud computing for scalable and sustainable education solutions," in AI-Enabled Sustainable Innovations in Education and Business. Hershey, PA, USA: IGI Global, 2025, pp. 149–170, doi: 10.4018/979-8-3373-3952-8.ch007.

[9] F. Bano et al., "Leveraging virtual reality in engineering education to optimize manufacturing sustainability in Industry 4.0," Sustainability, vol. 16, no. 18, p. 7927, 2024, doi: 10.3390/su16187927.

[10] S. F. Mallo et al., "A review on feasibility of web technology and cloud computing for sustainable ES: Leveraging AI, IoT, and security for green operations," J. Inf. Technol. Informatics, vol. 3, no. 2, 2024, doi: 10.31181/sems31202534a.

[11] M. Abaee, M. Saeedi, and H. Taherdoost, "Examining the role of IoT and cloud computing in achieving sustainable development goals," in Reshaping Environmental Science Through Machine Learning and IoT. Hershey, PA, USA: IGI Global, 2024, pp. 74–97, doi: 10.4018/979-8-3693-2351-9.ch005.

[12] K. J. Reddy and V. P. R. Pamaiahgari, "AI and data engineering transforming technical education for sustainability," in Integrating AI and Sustainability in TVET. Hershey, PA, USA: IGI Global, 2025, pp. 83–100, doi: 10.4018/979-8-3373-1142-5.ch004.

[13] U. Ibrahim, "The role of cloud computing in transforming ICT infrastructure in educational institutions," Int. J. Appl. Sci. Res., vol. 2, no. 2, pp. 213–226, 2024, doi: 10.59890/ijasr.v2i2.1333.

[14] N. Ahmad, N. Hoda, and F. Alahmari, "Developing a cloud-based mobile learning adoption model to promote sustainable education," Sustainability, vol. 12, no. 8, p. 3126, 2020, doi: 10.3390/su12083126.

[15] H. Sharma, E. Sengupta, and M. Saini, "Smart educational campuses: Leveraging AI and emerging technologies for sustainable development," in Proc. GINOTECH, 2025, pp. 1–6, doi: 10.1109/GINOTECH63460.2025.11076726.

[16] O. A. Oluokun et al., "Leveraging cloud computing and big data analytics for policy-driven energy optimization in smart cities," International Journal of Multidisciplinary Research and Growth Evaluation (IJMRGE), vol. 5, no. 1, pp. 1022–1034, 2024, doi: 10.54660/ijmrge.2024.5.1.1022-1034.

[17] D. D. Shallo, D. P. Sharma, and P. R. Chelliah, "Converging intelligence and energy efficiency in cloud environment for efficient resource allocation for sustainable SME growth," in Artificial Intelligence (AI) for IT Energy Efficiency and Green AI for Environment Sustainability, Cham, Switzerland: Springer Nature, 2026, pp. 589–620.

[18] A. AlSideiri, G. AlFarsi, and R. M. Tawafak, "Green computing in educational settings: Leveraging artificial intelligence," in Integrating AI and Security for Environmental Sustainability, Cham, Switzerland: Springer, 2025, pp. 235–242, doi: 10.1007/978-3-031-96641-5_21.

[19] N. S. Egbuhuzor et al., "Leveraging AI and cloud solutions for energy efficiency in large-scale manufacturing," Int. J. Sci. Res. Archive, vol. 13, no. 2, pp. 4170–4192, 2024, doi: 10.30574/ijsra.2024.13.2.2314.

[20] E. B. Estigoy et al., "Exploring intentions in leveraging cloud computing to transform learning experiences," Procedia Comput. Sci., vol. 272, pp. 131–138, 2025, doi: 10.1016/j.procs.2025.10.188.

[21] L. Harris, "The intersection of AI and cloud computing in green technology initiatives," Econ. Res.–Ekon. Istraživanja, vol. 33, pp. 1–18, 2024.

[22] P. S. Mehta et al., "Next-gen learning cloud integration for academic and industrial synergy," in Bridging Academia and Industry Through Cloud Integration, Hershey, PA, USA: IGI Global, 2025, pp. 73–108, doi: 10.4018/979-8-3693-6705-6.ch004.

[23] A. Atadoga et al., "Advancing green computing: Practices, strategies, and impact," World J. Adv. Eng. Technol. Sci., vol. 11, no. 1, pp. 220–230, 2024, doi: 10.30574/wjaets.2024.11.1.0052.

[24] A. Scrivano, "Innovative approaches in cloud computing," Authorea Preprints, 2025, doi: 10.36227/techrxiv.174970502.20312114/v2.

[25] A. Gautam and A. Kotiyal, "Connecting the future: Cloud-based IoT in education," in Integration of Cloud Computing and IoT. Boca Raton, FL, USA: Chapman & Hall/CRC, 2024, pp. 274–308, doi: 10.1201/9781032656694.

[26] S. C. Vetrivel et al., "Smart infrastructure for educational sustainability," in Smart Education and Sustainable Learning Environments in Smart Cities. Hershey, PA, USA: IGI Global, 2025, pp. 445–470, doi: 10.4018/979-8-3693-7723-9.

[27] X. Zang et al., "Data engineering for AI applications in energetic materials," FirePhysChem, 2025, doi: 10.1016/j.fpc.2025.11.010.

[28] A. Atadoga et al., "Tools, techniques, and trends in sustainable software engineering," World J. Adv. Eng. Technol. Sci., vol. 11, no. 1, pp. 231–239, 2024, doi: 10.30574/wjaets.2024.11.1.0051.

[29] H. S. Mahmood et al., "In-depth analysis of AI, IoT, web technology, and cloud computing for sustainable business," J. Inf. Technol. Informatics, vol. 3, no. 2, pp. 297–332, 2024, doi: 10.34218/IJCET_16_03_035.

[30] D. Chasokela and J. Hlongwane, "Leveraging AI and ICT for greener education," J. Res. Math. Sci. Technol. Educ., vol. 2, no. 1, pp. 36–45, 2025.

An Energy-Efficient and Modeling Attack-Resilient SNN-Driven PUF Security for Internet of Medical Things

Agila Harshini T, Harini Sriraman*
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

***Abstract*—The healthcare industry is rapidly evolving with the digitalisation of medical devices. Despite the advantages of IoMT devices, they remain vulnerable to cyberattacks that threaten patient data security. Authentication typically relies on a static key, making it susceptible mainly to modelling attacks. Existing techniques fail to balance security with lightweight operation for resource-constrained IoMT. A Physically Unclonable Function (PUF)-based technique provides genuine authentication with on-the-fly key generation but lacks resilience against modelling attacks. The proposed work presents a fusion of facial biometrics and a Spiking Neural Network(SNN) with a PUF. This fusion generates stronger entropy than conventional methods that rely on fixed stored keys. This work strengthens authentication by integrating user-specific facial biometrics, device-specific PUF responses, and SNN-induced nonlinearity. It enables joint authentication of both the device and the legitimate user. The experimental evaluation is conducted on the Digilent Basys3 Artix 7 FPGA board and achieves higher randomness in the generated responses, with a Shannon entropy of 0.693 nats and a normalized Hamming distance of 0.50. The security evaluation demonstrates that machine learning and deep learning attacks remain random-guess predictions, indicating that the proposed lightweight Face-SNN-PUF resists effective modelling of the challenge-response. Tailored for non-life-critical IoMT devices, the proposed resilient framework provides practical and energy-efficient authentication.**

Keywords*—PUF; SNN; biometric authentication; IoMT security; modeling attack

I. Introduction

The market for IoMT devices is growing, leading to easier monitoring of healthcare data through edge devices. To surveil and deliver healthcare to patients remotely, IoMT collects and transfers digital data to a server for processing. Despite the benefits, cybersecurity attacks are more prevalent among IoMT devices [2], [5], [13]. Healthcare data, when compromised by different attacks, can harm patients by disrupting personal data and causing financial loss or physical harm. The major security challenges are: Modeling attacks in which the attacker identifies and analyses the challenge-response pairs and creates a machine learning model to mimic the PUF behavior [4]; man-in-the-middle attacks that breach the device and enable reverse engineering or data extraction through physical access; authentication issues where the IoMT is vulnerable to cloning, spoofing, and replay attacks due to static or weak keys used in the device; and resource constraints, with the limited power and memory of small medical devices making them unsuitable to implement heavy cryptographic security [2], [11], [23]. Unlike other devices that cause only a data breach, IoMT leads to direct harm to patients. The existing PUF-based authentication models may remain vulnerable to modeling attacks when used without explicit binding between the user and the device. The proposed system combines three complementary entropy-contributing components: facial variability, randomness of spiking neurons, and PUF uniqueness to improve resistance against modeling attacks in IoMT authentication. Unlike conventional PUF-based authentication, the proposed work introduces a biologically inspired Neural Network to give temporal variability and non-linear challenge transformation. Fig. 1 shows the fusion model. The fusion of biometric features, spike dynamics, and hardware randomness increases the learning complexity of the challenge-response. The proposed work targets non-life-critical IoMT devices such as wearables and remote patient monitoring, where secure authentication with moderate latency is acceptable. Fig. 1 gives an illustration of the proposed fusion of the Face-SNN-PUF framework.

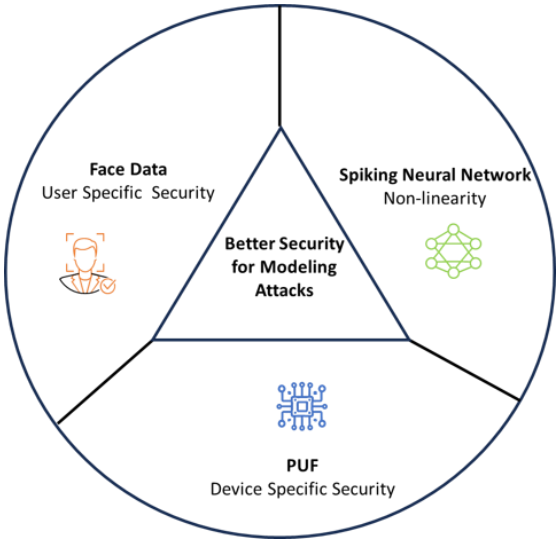


Fig. 1. Face-SNN-PUF fusion model.

The major contributions of this study are:

- A hybrid authentication framework integrating facial biometrics, Spiking Neural Networks, and Physically Unclonable Functions to ensure device and user authenticity.

*Corresponding author.

- The proposed Face-SNN-PUF enhances authentication security by combining biometric-derived entropy, non-linear transformation of SNN, and intrinsic PUF randomness, subsequently increasing resistance to modeling attacks.
- A detailed evaluation of security analysis against modeling attacks, using deep learning techniques, is provided.
- Lower computation and energy overhead compared to traditional cryptographic methods.

II. Related Work

To evaluate the proposed Face-SNN-PUF, the recent works in PUF-based authentication, biometric-assisted identity verification, and neuromorphic-based security are categorized and explored.

A. PUF-Based Authentication for IoMT

A stream cipher designed by a nonlinear feedback shift register is presented for data and device integrity by combining MAC and PUF response [1]. It is evaluated using NIST 800-22 and compared with existing ciphers using a heart disease dataset, and it presents a device authentication scheme using a hybrid oscillator arbiter PUF. PMsec authenticates the device before the data is read. It is suitable for low-power medical devices [2]. Based on blockchain and PUF, a scalable device authentication framework that achieves a uniqueness of 48.46% and reliability of 2.38% is proposed [3]. For role-based access, smart contracts are used. Ethereum's permissioned blockchain is developed as a proof of concept. To prevent the attacker from seizing the challenge bits, a protocol using a 64-bit arbiter PUF is presented that splits the challenge bits, and bit scrambling is also exploited by protocol extension. Also, the implications of modeling attacks by challenge patterns are studied [4].

PUF-based WMSNs protocol, a fuzzy extractor, a one-way hash function, and XOR operations are used for security [5]. Compared to other schemes, it requires low communication and computation costs. The proposed integration of PUF with IOTA and blockchain uses the Masked authentication message protocol to store and share data. With minimal energy requirements, decentralized access control is provided [6]. To authenticate the IoT sensor nodes, a PUF-based lightweight mechanism is proposed. The security resists network attacks and has low computational requirements and higher throughput [7]. A lightweight GAKA that uses PUF to solve secure transmission and data storage is given. It has higher computational efficiency than other systems [8]. The random variations in the PUF are affected by external factors. The proposed design has increased performance, reliability, and uniformity [9]. For resource-constrained devices, a RoT architecture based on PUF is proposed. It is robust and cost-efficient and uses SRAM and a syndrome-based fuzzy extractor, reinforcing device-centric security [10].

Combining the fingerprint of the device with an authentication protocol, a PUF-based identity framework for IoT devices is proposed. It is an identity management system at the IoT gateway layer with low latency. It has resistance to impersonation and replay attacks [11]. To support dynamic response generation, a configurable C-PUF is introduced. Demonstrates higher entropy and stability under varying environmental conditions [12]. Focusing on COVID-19 patient monitoring, a PUF-assisted key protocol for medical IoT is developed. With reduced computational cost, it uses device-generated randomness and mutual authentication [13]. A reconfigurable PUF-based authentication for healthcare IoT is proposed that reconfigures challenge-response and resists modeling attacks. Enhances authentication reliability and secure communication [14]. Ensuring secure key agreement in sensor, gateway, and cloud, a three-layer authentication protocol using PUF is presented. The authentication is verified with formal and simulation-based methods [15].

A PUF-based anonymous user authentication for IoT devices is developed. Balances anonymity and traceability through cryptographic primitives with high scalability and resistance to replay attacks [16]. Introduces memory-based Mc-PUF, utilising SRAM and DRAM timing variation. With high uniqueness and stability, it resists machine learning and modeling attacks [17]. Proposes a feedback-based PUF for resisting ML attacks. Provides a nonlinear challenge-response by using response-feedback loops. Shows strong resistance to logistic regression and SVM modeling [18]. Introduces a bias-independent PUF that applies delay calibration and symmetric routing for uniform response distribution. It is resistant to layout-induced imbalance [19]. A priority arbiter PUF that enhances a standard arbiter PUF is proposed. To improve delay sensitivity and uniqueness, a priority encoder is used. Achieves strong resistance to ML and environmental noise [20].

B. Biometric Integrated Approaches

The proposed neighbor-assisted, PUF-based three-factor authentication. 100% reliability is demonstrated [21]. The IoHT are vulnerable to insider threats and physical attacks. Mutual authentication with biometrics and PUF is proposed. The system is tested with automated validation of Internet security protocols [22]. A robust identity verification in medical sensors, a two-factor authentication using PUF and password, is designed. It has enhanced resistance to side-channel and impersonation attacks [23]. A combination of PUF with biometrics for hybrid authentication is explored for stronger identity. Highlights improved entropy [24].

C. Neuromorphic and SNN-Based Security

A spiking neural network-based framework for privacy in IoMT in the metaverse is presented. Neuron behaviour for adaptive and energy-efficient secure health data exchange is given [25]. A single spiking neuron model using membrane potential dynamics for an entropy source is proposed. Demonstrates strong behaviour of PUF with minimal hardware cost [26]. Proving that biological neuron firing can give a unique PUF response, implements neuron-spiked circuits to generate challenge-dependent outputs [27].

The above related studies primarily address user or device authentication separately. Existing PUFs have improved security by increasing the complexity of the architecture, XOR structures, feed-forward paths, or protocol-level protection. These approaches have fixed challenge-response behavior that introduces hardware overhead and computational complexity.

Primarily, few methods restrict CRP exposure, while others improve resistance to specific machine-learning models. The reviewed methods do not simultaneously address user device binding and energy awareness for IoMT devices. The proposed Face-SNN-PUF addresses this gap with modelling-attack resilience and user device binding. Table I gives a comparison of other authentication approaches to highlight the design objectives of the proposed framework.

TABLE I. COMPARISON OF IOMT AUTHENTICATION AND SECURITY FRAMEWORKS

S.No	Security Metric	Biometric-only system	PUF-only System	Proposed Face Biometric–SNN–PUF
1	Resistance to Spoofing	High (with fake data)	Strong (Unique to device, no user binding)	Strong(User biometric + PUF must match)
2	Resistance to ML and DL Modeling	Vulnerable	Weak (PUF can be modeled)	Improved resistance by SNN-induced nonlinearity (ML/DL results are given in Section IV)
3	Cloning Resistance	Weak (Biometrics can be copied)	Strong(Unique per chip)	Strong (Requires Match in User device domain)
4	Lightweight Efficiency	Moderate (needs storage)	Lightweight	Lightweight (PUF+SNN power ~0.004 W)
5	Entropy	Limited (~6-8 bits)	Arbiter PUF (~10-12 bits)	Spike+ PUF (~13-14 bits)

III. PROPOSED FRAMEWORK OF FACE-SNN-PUF

The proposed work combines the biometric (face), Spiking Neural Network (SNN), and arbiter PUF, providing lightweight and secure authentication for IoMT devices. 10000 CRPs are collected as a dataset from the UART connected to the board for analysis. Fig. 2 shows the architecture of the proposed Face-SNN-PUF system.

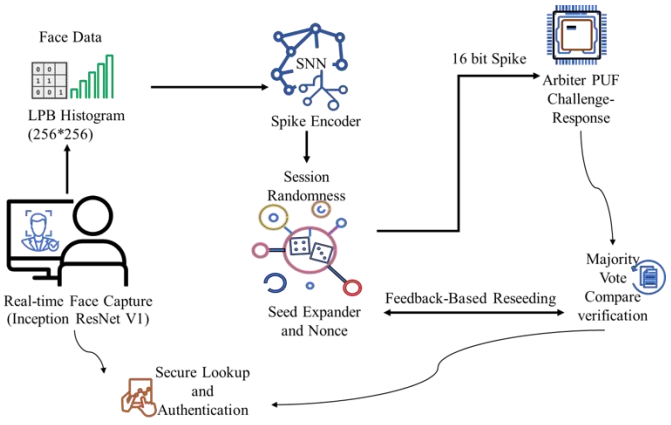


Fig. 2. Architecture of the Face-SNN-PUF system.

1) The biometric data is captured from the user in real-time. The facial images of ten users are captured and registered for authentication. Genuine authentication was evaluated using samples from the corresponding registered user. The samples from other users were used to evaluate imposter authentication attempts. Using a pre-trained InceptionResNetV1 via FaceNet, deep embeddings are extracted. Separately, the detected facial region is processed using lightweight Local Binary Pattern (LBP) extraction. The lightweight LBP(Local Binary Pattern) compares each pixel with 8 neighbours to form an 8-bit binary number(0-255). The LBP histogram counts the number of times each LBP code occurs. Each histogram bin value is multiplied by its corresponding index and summed to produce a compact biometric-derived value. For reproducibility, the LBP uses a radius of 1 pixel with 8 sampling points. The 16-bit final face signature is obtained by applying a modulo 2^{16} operation on the weighted sum. The bitmask keeps only the last 16 bits of the sum to ensure compactness for the FPGA; this 16-bit value is used as a compact biometric-derived input to the hardware authentication pipeline, whereas biometric recognition is performed using the InceptionResNetV1 embedding similarity. A 16-bit width can efficiently fit into the Basys 3 FPGA board. To examine the inter-person variation of the compact representation, the LBP-based face generation procedure was evaluated using ten different facial samples.

2) The SNN(biologically inspired spike-based) reads the person identifier (PID) derived from the face signature. Then selects one of the thresholds for that PID. To provide a source of variability for repeated identical person ID, a free-running jitter counter is added. The extracted features are encoded into spike trains by an SNN encoder module. The spiking variability and threshold selection are clock-edge jitter-influenced by the 100 MHz on-board oscillator and metastable delay variations. The SNN engages a leaky integrate-and-fire neuron model with discrete threshold levels from a predefined range [Tmin, Tmax]. The effective threshold is chosen dynamically based on the PID and jitter count value.

3) To provide non-linearity, the spike generator maps a person's identity and features into the spiking activity (spike 16). To introduce session-level variation, an 8-bit free-running nonce counter is implemented on the FPGA. The nonce is reset to zero and incremented at each rising edge of the 100 MHz system clock. The seed expander combines the biometric-derived input, selected SNN threshold, and the current 8-bit nonce to generate the 16-bit PUF seed. The nonce is generated internally by the FPGA and is not transmitted by the host application; therefore, no separate nonce-exchange or synchronization message is required between the Python application and the FPGA. The 16-bit arbiter PUF with LFSR-driven challenges is seeded with the input. The arbiter PUF consists of 16 delay stages, and challenges are generated using a 16-bit linear feedback shift register (LFSR) with a maximal-length primitive polynomial. The PUF gives a device-specific challenge-response pair. To improve the stability of the response, each challenge is evaluated multiple times and a majority voting scheme is applied across repeated measurements. The majority of responses are stored in a secure lookup table. During

enrollment, reference responses are recorded for a fixed set of challenges. During authentication, live responses are matched against the enrolled references. During authentication, the live facial embedding is compared with the enrolled user using the cosine similarity. A minimum threshold of 0.60 is used, with the per-user threshold determined from the mean enrollment similarity minus a margin of 0.05. Authentication requires consistent recognition for three consecutive frames for access to be granted. For validation of the attack analysis, 10,000 challenge-response pairs (CRPs) are collected over UART from the FPGA.

For reproducibility, the key design and implementation parameters of the proposed Face-SNN-PUF are summarized in Table II.

TABLE II. KEY IMPLEMENTATION PARAMETERS OF THE PROPOSED FRAMEWORK

Module	Key parameter	Description
Hardware platform	FPGA board	Xilinx Basys 3 (Artix-7 XC7A35T-1CPG236C
Development environment	Tool	Xilinx Vivado
Biometric pursuit	Face embedding model	InceptionResNetV1 (FaceNet, pre-trained)
Feature reduction	Local Binary Pattern (LBP)	Radius 1, Neighbors 8
Face signature	Dimensionality reduction	LBP histogram weighted sum, reduced to 16 bits using modulo 216
Neuromorphic encoding	SNN neuron model	Leaky Integrate-and-Fire (LIF)
SNN variability source	Threshold selection	PID-dependent threshold with jitter-based variation
PUF architecture	Arbiter PUF	16-bit arbiter PUF
Challenge generation	LFSR configuration	16-bit LFSR
Seed construction	Entropy fusion	Concatenation of spike output, threshold index, and session nonce
Response stabilization	Noise mitigation	Majority voting over repeated evaluations
Scalability	Design extensibility	Extensible to higher PUF bit-widths with resource trade-offs

B. Attack Model

The attacker is assumed to have access to collected CRPs and may attempt modeling attacks. Potentially stronger capabilities, such as access to captured facial images, inferred PID information, or observed intermediate signals, are recognized as possible attack surfaces of the proposed framework. To examine the robustness of the modeling-attack evaluation, the experiments were repeated using five different random seeds. Logistic Regression, Random Forest, and MLP achieved mean prediction accuracies of 48.63%, 49.49%, and 49.67%, respectively, with standard deviations of 1.15%, 0.69%, and 1.25%. The corresponding mean ROC-AUC values were 0.481, 0.498, and 0.493, respectively. The target classes contained 5056 zero samples and 4944 one samples, indicating a nearly balanced distribution. In addition, the maximum absolute Pearson correlation between an individual challenge bit and the target was approximately 0.0131, while the maximum observed mutual information was 8.55×10^{-5}bits. These results demonstrate that the evaluated models remained close to the random-guessing baseline and did not identify a strong predictive relationship in the evaluated CRP data.

IV. RESULTS AND ANALYSIS

The biometric template and PUF keys are not stored. Reproducing this behaviour is hard, even with full access to the device. The proposed Face-SNN-PUF has favourable entropy, uniqueness, and strong resistance to DL-based modeling attacks. Fig. 3 represents the randomness of the proposed model PUF keys. The values are centred around 8, which shows a balanced distribution of 1s and 0s across the collected samples. Fig. 4 shows the uniqueness comparison of the PUF with existing works [3], [4], [22], [25].

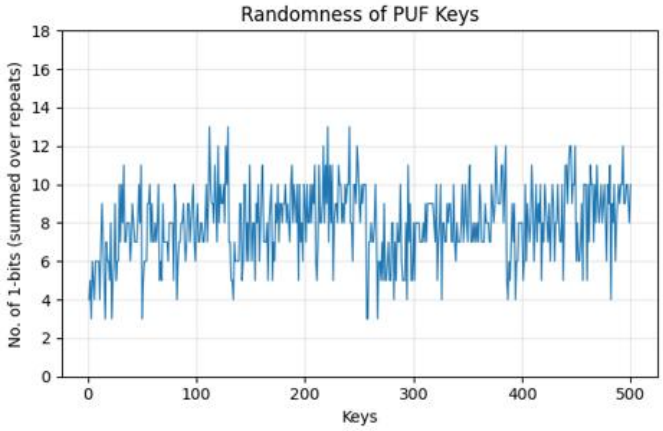


Fig. 3. Randomness analysis of the PUF keys.

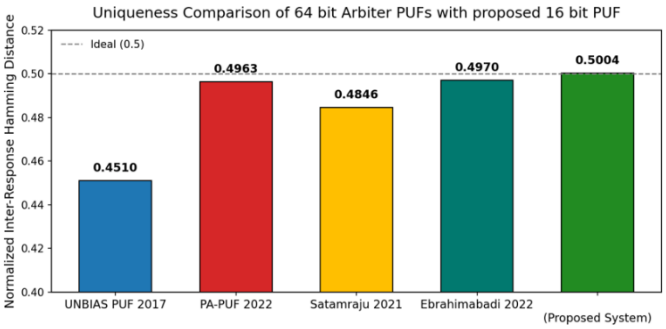


Fig. 4. Comparison of uniqueness of the PUF keys.

A. Randomness and Uniqueness Analysis

The Shannon entropy and inter-response Hamming distance measures are analysed for the Bio-SNN-PUF to find the randomness and uniqueness. The measured Shannon entropy is 0.693, representing near-ideal randomness for a binary response distribution. This corresponds to the theoretical maximum entropy of a balanced binary distribution, indicating the generated responses exhibit near-ideal randomness. The average normalised inter-response Hamming distance is 0.5004, and the given challenges differ by approximately 50%. This shows that the PUF has strong uniqueness. Fig. 5 shows the quantitative analysis of the Bio-SNN-PUF's randomness and uniqueness. The samples produced ten different 16-bit values with no collision. The mean pairwise Hamming distance

between the generated values was 7.13 bits out of 16, with a minimum of 3 bits and a maximum of 11 bits. The overall proportion of 1-bits was 48.13%, indicating a reasonably balanced distribution of the generated compact values. These results provide an initial empirical assessment of inter-person variation.

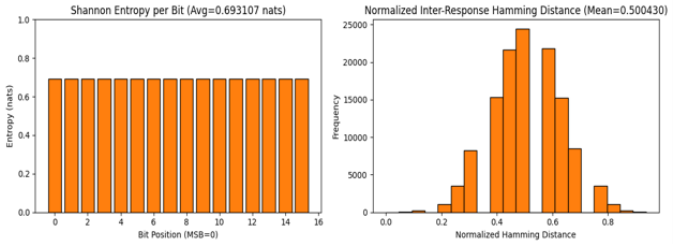


Fig. 5. Entropy and uniqueness of Face-SNN-PUF.

B. Entropy Analysis of SNN-Assisted

The randomness of spike 16 is evaluated by computing Shannon entropy (a measure of average randomness in 16-bit output), which gave 13.14 bits out of 16, and min-entropy (the worst-case predictability is captured) of 11.29 bits. This shows the response space is highly random. Even under worst-case conditions, the spike output retains entropy. The average per-bit entropy is 0.9999(good spread of 0s and 1s), which is nearly 16 independent effective bits. With the PUF having a unique response per device, the user-dependent biometric input and the spiking transformations increase the randomness. Fig. 6 shows the comparison to the existing works [20], [21]; the proposed model has higher entropy. The facial biometric, SNN, and PUF components are coupled. The face input determines the person identifier, which influences the SNN spike and threshold. These are combined with the session nonce to construct the PUF seed.

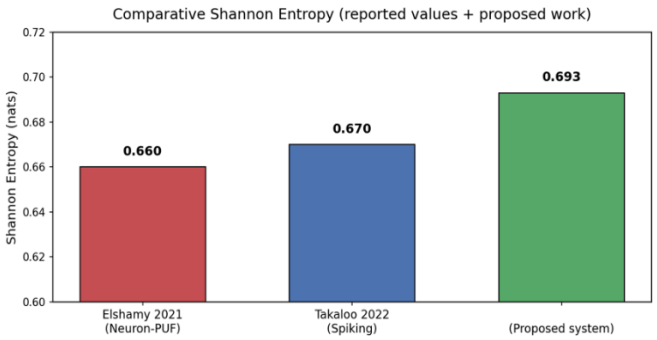


Fig. 6. Comparison of the entropy of the face-SNN-PUF.

C. Uniqueness and Reliability Analysis

The inter-challenge stability, the probability that two different challenges yield the same PUF response, is measured to characterize the uniqueness of generated responses. The average identical response is 52.6%, which is the inter-challenge collision rate. The intra-challenge Hamming distance is 1.52 bits out of 16. It is a measure of variation when the same challenges are repeated multiple times. It indicates the response consistency under repeated evaluations. This exhibits a moderate noise level but remains within the range of stochastic PUF designs.

D. Security Analysis

The security of the proposed work, Biometric + SNN + PUF, provides improved resistance to modeling compared to biometric-only or PUF-only approaches. It is more resistant to modeling and cloning. This balances strong security and lightweight efficiency that is practical for real-world security of IoMT devices.

A biometric-only or PUF-only system is weak and can be modelled. The proposed method introduces a nonlinear biometric input and the entropy of Spike patterns, which makes the approximation of the ML and DL model harder. The biometric data of each person gets a unique PUF instance through dynamic seed expansion. The spike latency, firing rate, and threshold randomness of the SNN introduce nonlinearity. The spike entropy ensures that the input-output correlation remains nonlinear and is highly device-specific, so the attacker cannot use an ML model to approximate it easily. The modeling is a supervised attack; the attacker collects the CRPs. 10000 CRPs are used, training=8000 and testing = 2000. The PUF seed is generated by the FPGA seed-expander logic using the biometric-derived input, selected SNN threshold, and the 8-bit free-running nonce counter. These features are used to train classifiers to predict responses for unseen challenges. The response prediction represents a binary classification problem, so an attacker without meaningful knowledge about PUF behaviour can only achieve 50% prediction accuracy, which is random guessing. The DL models are also ~50%, which states that the models cannot learn the pattern and an attacker fails to approximate the challenge-response relationship.

E. Resistance to Machine Learning-Based Modeling Attacks

The proposed system is evaluated for resilience against three ML-based classifiers. The accuracy achieved by Logistic Regression is 48.0%, Support Vector Machine 50.3%, and MLP 51.2%. Fig. 7 represents these accuracies. The results indicate that all machine learning models exhibit a random guess line prediction performance, which limits the attacker's ability to learn the challenge-response behavior.

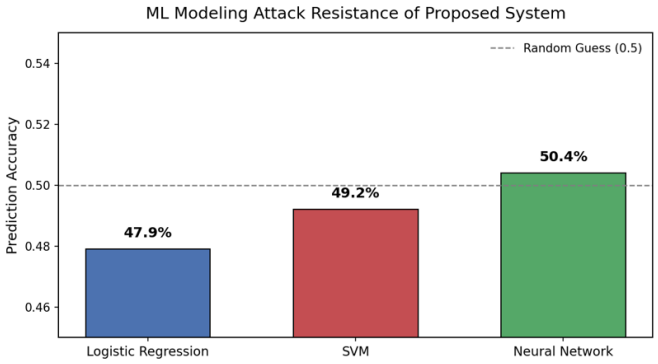


Fig. 7. Prediction performance of machine learning-based modeling attacks (Lower the deviation of random guessing, higher the security).

F. Resistance to Deep Learning-Based Modeling Attacks

Further evaluation of resilience against sophisticated modeling attacks: a DNN consisting of three layers(64-32-16 neurons and ReLU activation function) and an advanced deep learning model that uses a 1D-CNN front-end, residual blocks,

and batch normalisation are trained using the collected CRPs. The accuracies learned are 50.2% and 51.75%. The results in Fig. 8 show that the deep neural models prevent effective approximation of the challenge response of the Bio-SNN-PUF.

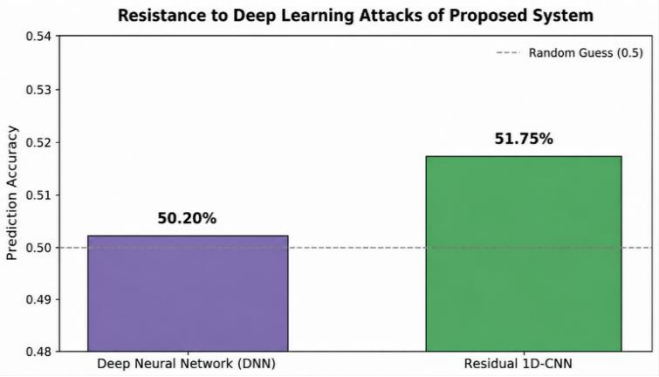


Fig. 8. Prediction performance of deep learning-based modeling attacks (Values close to random guessing indicate higher security).

G. Statistical Analysis of Modeling Attack Results

A systematic evaluation of modeling attacks is done using five models with five independent train-test splits. The Mean accuracy, Standard Deviation,95% Confidence level, and Mean ROC0AUC are calculated as shown in Table III. The results remain close to 50%, indicating the models did not obtain a consistent relationship. This means that none of the evaluated models was able to learn the response pattern. This consistency of results using ML and DL models supports modeling resistance.

TABLE III. STATISTICAL EVALUATION OF MODELING ATTACK

Model	Mean Accuracy	SD %	95%CI%	Mean ROC-AUC
Logistic Regression	49.51	0.63	48.73-50.29	0.491
SVM	49.80	1.36	48.11-51.49	0.502
MLP	49.95	2.02	47.45- 52.45	0.495
DNN	50.52	1.12	49.13-51.91	0.509
Residual 1D-CNN	49.66	1.40	47.92-51.40	0.490

H. Comparison of ML and DL Attack Resistance with Existing Works

The biometric and the SNN are encoded as a challenge, which will confuse the DL models. The attack resistance of the proposed system compared with the existing solutions [4], [9], [14], [18] demonstrates improved modeling resistance. Fig. 9 shows the prediction accuracy comparison with the existing work.

I. Cloning Resistance

The proposed work resists cloning because it requires the biometric details, the spike nonlinearity, and the FPGA data, which makes cloning practically infeasible. For a cloning attack to be successful, an adversary would need to reproduce the combined effects of the facial input, spike dynamics, and device-specific PUF, significantly increasing the difficulty of device cloning. The attacker requires multiple independent entropy sources for successful impersonation of user-specific biometrics, SNN spike patterns, and PUF variation for cloning.

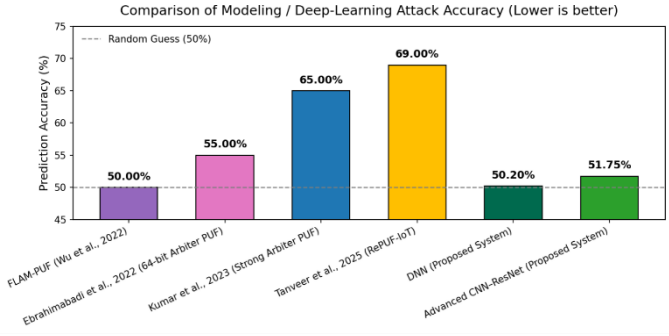


Fig. 9. Comparison of the DL attack resistance of the Face-SNN-PUF.

V. ENERGY EFFICIENCY

The energy efficiency of the proposed method exhibits low energy consumption, making it suitable for resource-constrained IoMT devices. Fig. 10 shows the power report of the implemented Face-SNN-PUF. The existing authentication methods [1], [2], [3] based on PUF, which primarily focus on security and consume high power, and the multifactor key authentication [4], [5], [26], although reliable, are less suitable for low-power devices. Neuron-inspired architectures [20], [21] focus on entropy generation but lack FPGA-level implementation. Compared to existing methods, the proposed work, implemented on an FPGA, offers reduced energy consumption, providing entropy and modeling attack resilience. Fig. 10 shows the measured power consumption of the proposed work implemented on the Basys 3 FPGA. The proposed Face-SNN-PUF operates at a clock frequency of 100 MHz, corresponding to a clock period of 10ns.

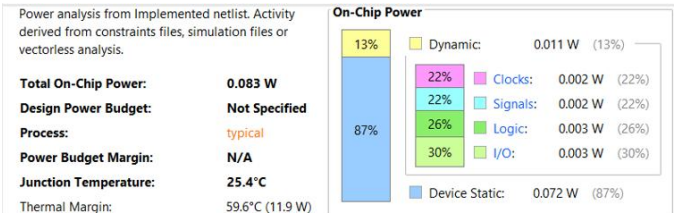


Fig. 10. Power report of the Face-SNN-PUF.

A. Energy Per Authentication

The energy consumption is calculated as,

$$E_{auth} = P_{dynamic} * T_{auth}$$

From the RTL cycle analysis, a complete authentication requires 294 clock cycles, including the 256 -cycle SNN processing window, PUF evaluations, and majority voting.

Therefore, the estimated latency is,

$$T_{auth} = 294 * 10ns = 2.94\ \mu s$$

The Vivado power analysis PUF+SNN estimates dynamic power consumption at 0.004W. SO the estimated energy per authentication is,

$$E_{auth} = 0.004 * 2.94 * 10\text{-}6 = 11.76nJ$$

The energy value is estimated from the Vivado power analysis and RTL-derived authentication latency.

VI. Conclusion and Future Work

The entropy is derived from three coupled components: the face biometrics, spiking neural nonlinearity, and the PUF. Their fusion is an effective security mechanism for improving ML and DL modeling resistance. It is suitable for resource-constrained IoMT devices due to its minimal FPGA usage. Compared to a conventional PUF-only or a biometric-only method, the proposed system transforms the biometric nonlinearly, making modeling attacks harder. There is no need for key storage, which offers resilience against spoofing and Man-in-the-Middle attacks. This facial biometric neuromorphic PUF security contributes towards the development of resilient and intelligent IoMT devices. Although the proposed work shows increased resistance against modeling attacks, the study has been evaluated on a single hardware platform and primarily for non-life-critical IoMT devices. Future investigations will consider multiple FPGA instances and environmental robustness. The future work will focus on extending the PUF to higher bit-widths for enhanced entropy, fusion of dynamic biometrics (ECG, iris) for different healthcare scenarios, and real-time deployment on wearable IoMT devices under practical operating conditions.

Availability of Data and Materials

The code and dataset used in the research are available at the GitHub link https://github.com/agilaharshini/Bio-SNN-PUF.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not- for-profit sectors.

Competing Interest

The authors (Dr. Harini S and Agilaharshini T) declare that there are no conflicts of interest related to this manuscript. No financial, professional, or personal relationships with organizations or individuals have influenced the research, analysis, or conclusions presented in this work.

Acknowledgment

The authors would like to take this opportunity to thank the management of Vellore Institute of Technology for providing the APC and encouragement to carry out this research.

References

[1] Punam Kumari, Bhaskar Mondal, “Lightweight encryption with data and device integrity using NLFSR and PUF for the Internet of Medical Things”,Internet of Things, Volume25,2024,101041,ISSN,25426605,https://doi.org/10.1016/j.iot.2023.101041.

[2] V. P. Yanambaka, S. P. Mohanty, E. Kougianos and D. Puthal, "PMsec: Physical Unclonable Function-Based Robust and Lightweight Authentication in the Internet of Medical Things," in IEEE Transactions on Consumer Electronics, vol. 65, no. 3, pp. 388-397, Aug. 2019, doi: 10.1109/TCE.2019.2926192.

[3] Krishna Prasad Satamraju, B. Malarkodi,A decentralized framework for device authentication and data security in the next generation internet of medical things,Computer Communications,Volume 180,2021,Pages 146-160,ISSN0140-3664,https://doi.org/10.1016/j.comcom.2021.09.012.

[4] M. Ebrahimabadi, M. Younis and N. Karimi, "A PUF-Based Modeling-Attack Resilient Authentication Protocol for IoT Devices," in IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3684-3703, 1 March1, 2022, doi: 10.1109/JIOT.2021.3098496.

[5] Shao, X., Guo, Y. & Guo, Y. A PUF-based anonymous authentication protocol for wireless medical sensor networks. Wireless Networks 28, 3753–3770 (2022). https://doi.org/10.1007/s11276-022-03070-1

[6] Bathalapalli VKVV, Mohanty SP, Kougianos E, Iyer V, Rout B. PUFchain 3.0: Hardware-Assisted Distributed Ledger for Robust Authentication in Healthcare Cyber–Physical Systems. Sensors. 2024; 24(3):938. https://doi.org/10.3390/s24030938

[7] Kumar Chaudhary, R.R., Chatterjee, K. A Lightweight PUF based Multi-factor Authentication Technique for Intelligent Smart Healthcare System. Peer-to-Peer Netw. Appl. 16, 1975–1992 (2023). https://doi.org/10.1007/s12083-023-01509-6

[8] T. -F. Lee, X. Ye and S. -H. Lin, "Anonymous Dynamic Group Authenticated Key Agreements Using Physical Unclonable Functions for Internet of Medical Things," in IEEE Internet of Things Journal, vol. 9, no. 16, pp. 15336-15348, 15 Aug.15, 2022, doi: 10.1109/JIOT.2022.3149117.

[9] A. Kumar, K. Chatterjee and B. Mondal, "A Strong PUF-based Authentication for Medical IoT," 2023 IEEE 2nd International Conference on Industrial Electronics: Developments & Applications (ICIDeA), Imphal, India, 2023, pp. 281-285, doi: 10.1109/ICIDeA59866.2023.10295174.

[10] C. Lee, M. Oh, D. Kwon, Y. Park and Y. Park, "PLAKA-MD: PUF-Based Lightweight Authentication and Key Agreement Scheme for Medical Devices in IoHT," in IEEE Internet of Things Journal, vol. 12, no. 12, pp. 19338-19353, 15 June15 2025, doi: 10.1109/JIOT.2025.3543488.

[11] Huang, Z., Wang, Q. A PUF-based unified identity verification framework for secure IoT hardware via device authentication. World Wide Web 23, 1057–1088 (2020). https://doi.org/10.1007/s11280-019-00677-x

[12] F. Amsaad et al., "Enhancing the Performance of Lightweight Configurable PUF for Robust IoT Hardware-Assisted Security," in IEEE Access, vol. 9, pp. 136792-136810, 2021, doi: 10.1109/ACCESS.2021.3117240.

[13] M. Masud et al., "A Lightweight and Robust Secure Key Establishment Protocol for Internet of Medical Things in COVID-19 Patients Care," in IEEE Internet of Things Journal, vol. 8, no. 21, pp. 15694-15703, 1 Nov.1, 2021, doi: 10.1109/JIOT.2020.3047662.

[14] M. Tanveer, A. G. Alharbi and S. A. Aldossari, "RePUF-IoT: Reconfigurable PUF-Based Authentication Protocol for IoT-Driven Healthcare Systems," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2025.3602311.

[15] A. M. Aminian Modarres, N. S. Anzabi-Nezhad and M. Zare, "A New PUF-Based Protocol for Mutual Authentication and Key Agreement Between Three Layers of Entities in Cloud-Based IoMT Networks," in IEEE Access, vol. 12, pp. 21807-21824, 2024, doi: 10.1109/ACCESS.2024.3361921.

[16] W. Wang and F. Wen, "A Robust Two-factor Medical Sensor Authentication Protocol Based on PUF," 2024 International Seminar on Artificial Intelligence, Computer Technology and Control Engineering (ACTCE), Wuhan, China, 2024, pp. 462-466, doi: 10.1109/ACTCE65085.2024.00100.

[17] Kavya Dayananda, Nima Karimian, When Physical Unclonable Function Meets Biometrics,arXiv, 14 Dec 2020

[18] L. Wu, Y. Hu, K. Zhang, W. Li, X. Xu and W. Chang, "FLAM-PUF: A Response–Feedback-Based Lightweight Anti-Machine-Learning-Attack PUF," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 11, pp. 4433-4444, Nov. 2022, doi: 10.1109/TCAD.2022.3197696.

[19] S. A. Khowaja, K. Dahri, M. A. Jarwar and I. H. Lee, "Spike learning based Privacy Preservation of Internet of Medical Things in Metaverse," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2023.3306704.

[20] M. Elshamy and H. -G. Stratigopoulos, "Neuron-PUF: Physical Unclonable Function Based on a Single Spiking Neuron," 2021 IEEE

27th International Symposium on On-Line Testing and Robust System Design (IOLTS), Torino, Italy, 2021, pp. 1-6, doi: 10.1109/IOLTS52814.2021.9486716.

[21] H. Takaloo, M. Ahmadi and A. Ahmadi, "Spiking Neurons: A New Entropy Source for Physically Unclonable Functions," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 170-174, doi: 10.1109/ISCAS48785.2022.9937747.

[22] Wang, W. C., Li, Z., Skudlarek, J., Larouche, M., Chen, M., & Gupta, P. (2017). UNBIAS PUF: A Physical Implementation Bias Agnostic Strong PUF. arXiv preprint arXiv:1703.10725.

[23] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. P. C. Rodrigues and Y. Park, "Physically Secure Lightweight Anonymous User Authentication Protocol for Internet of Things Using Physically Unclonable Functions," in IEEE Access, vol. 7, pp. 85627-85644, 2019, doi: 10.1109/ACCESS.2019.2926578.

[24] P. Williams, H. Idriss and M. Bayoumi, "Mc-PUF: Memory-based and Machine Learning Resilient Strong PUF for Device Authentication in Internet of Things," 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 2021, pp. 61-65, doi: 10.1109/CSR51186.2021.9527930.

[25] Singh, S., Bodapati, S., Patkar, S., Leupers, R., Chattopadhyay, A., & Merchant, F. (2022, October). Pa-puf: A novel priority arbiter puf. In 2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC) (pp. 1-6). IEEE.

[26] P. K. Sadhu, V. P. Yanambaka, A. Abdelgawad and K. Yelamarthi, "NAHAP: PUF-Based Three Factor Authentication System for Internet of Medical Things," in IEEE Consumer Electronics Magazine, vol. 12, no. 3, pp. 107-115, 1 May 2023, doi: 10.1109/MCE.2022.3176420.

[27] E. -N. Sassalou et al., "A PUF-based Root-of-Trust for resource-constrained IoT devices," 2025 IEEE International Conference on Cyber Security and Resilience (CSR), Chania, Crete, Greece, 2025, pp. 824-831, doi: 10.1109/CSR64739.2025.11130114.

A Machine Learning Framework for Wheat Seed Variety Classification Using GLCM-Based Texture Analysis and Feature Optimization

Muhammad Munawar Ahmed[1], Malik Muhammad Saad Missen[2],
Hannan Adeel[3], Ahamad Zaki Bin Mohamed Noor[4], Muzamil Malik[5]
Department of Information Technology, The Islamia University of Bahawalpur, Bahawalpur, Pakistan[1]
Department of Software Engineering, The Islamia University of Bahawalpur, Bahawalpur, Pakistan[2]
Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Malaysia[3, 4]
Department of Computing, Hamdard University, Islamabad, Pakistan[5]

***Abstract*—In Pakistan, wheat is one of the important crops, and the production of wheat is highly reliant upon the use of pure and certified seed varieties. Hence, wheat seeds should be properly identified to ensure the quality of the crop and better production. Farmers typically rely on a visual assessment of the varieties of seeds in practice. This technique is incorrect and prone to inaccuracies. Many seeds of wheat are very similar in shape and color. This wrong seed selection could lead to reduced yield and economic loss. To tackle this problem, this study proposes an automated machine learning based classification of wheat seed varieties. The experiments are carried out on six wheat varieties grown in Southern Punjab, Pakistan. This research utilizes the data obtained by taking digital images of wheat seeds under consistent lighting and a uniform background at three different resolutions: 256 x 256, 512 x 512, and 1024 x 1024. In seed analysis, texture is chosen to describe seed surface characteristics because texture information is best at differentiating visually similar seeds. These images are used to calculate the following features: Histogram Features, Texture Features, and Laws Texture Features using Spatial Gray Level Co-occurrence Matrices. These matrices are second-order statistics such as correlation, energy, contrast, and homogeneity calculated based on the relationship between the intensities of the pixels at various distances and orientations. Then, the derived features are optimized using Principal Component Analysis and Correlation Feature Selection. Five machine learning classifiers, namely LogitBoost (LB), BayesNet (BN), Classification Via Regression (CVR), Multi-Layer Perceptron (MLP), and Random Forest (RF), are employed. The classifiers are evaluated separately on the PCA-reduced and CFS-selected datasets for each image resolution. Classification accuracy and consistency were assessed. The experimental results indicated that the PCA optimized Dataset of 1024 x 1024 resolution and the Random Forest classifier gave excellent discrimination between the wheat varieties, and the classifiers achieved 96.4% accuracy for all the classes. The proposed system helps to minimize human judgment dependency and ultimately reduces the risk of misclassification at the early stage of cultivation. The method is non-destructive, economical. and applicable in agricultural applications.**

***Keywords*—Gray-Level Co-occurrence Matrix (GLCM); Principal Component Analysis (PCA); Correlation-based Feature Selection (CFS); Random Forest (RF); BayesNet (BN)**

I. INTRODUCTION

Wheat (Triticum aestivum L.) is an important cereal crop in the world and is a source of food for a large portion of the human population. Wheat is important in food production because of its versatility across different agro-climates, high nutritional content, and its broad use in foods, making it vital to global food security [1, 2]. In Pakistan, wheat is a major staple crop which plays a crucial role in rural livelihoods, agricultural productivity, and Pakistan's economic development. Wheat production has significantly improved due to the introduction of high-yielding wheat varieties, increased irrigation, use of fertilizers, and the use of modern cultivation methods since the Green Revolution, which has contributed to food security in the country [3]. However, the sustainability of wheat production is becoming a growing concern due to climate change, water scarcity, soil degradation, increasing temperatures, and increased occurrence of biotic stresses (such as pests and diseases) and abiotic stresses (such as drought, heat, and salinity) [4, 5].

In response to these difficulties, agricultural research stations have cultivated many wheat cultivars with greater yield capacity, disease resistance, drought resistance, and suitability to various environmental conditions [6]. As a result, a few wheat varieties have been grown throughout Pakistan, especially in the South of Punjab, where wheat farming is a critical part of the agricultural activities [7]. Careful choice of wheat cultivars is very important as they have different agronomic characteristics, growth requirements, stress tolerances, and yield potentials. Therefore, correct wheat variety identification and classification is crucial in agricultural production, varietal purity testing, wheat breeding, seed certification, and precision agriculture [8, 9].

The identification of varieties is traditionally done with the help of experienced wheat agronomists and seed analysts by manual inspection. These methods tend to be time-consuming, labour-intensive, subjective, and subject to human error, especially when differentiating morphologically similar cultivars [10, 11].

In recent years, there has been a growing demand for fast, accurate, and scalable classification approaches, which has encouraged the use of computer vision, digital image processing, and machine learning technologies in agricultural systems. These technologies can be used to automatically analyze the visual features of grains, including morphology, texture, color, shape, and structural characteristics. These features can help discriminate between different wheat varieties [12, 13].

Machine vision systems use vision equipment, sensors, and computational algorithms to accurately and consistently acquire and analyze visual information [14]. Machine vision systems can analyze a vast amount of data without getting tired or biased, which is very suitable for agricultural automation [11, 15]. This capability extends to image processing, where digital images are processed using image enhancement, image segmentation, feature extraction, and pattern recognition to transform the raw images into meaningful analytical information [13, 16]. The quantitative attributes derived from extracted features can be used by machine learning algorithms for classification and decision-making.

Over the past years, artificial intelligence has been used in crop classification in a way that has greatly enhanced its effectiveness. In recent years, machine learning models like Support Vector Machines (SVMs), ensemble classifiers, and deep learning models like Convolutional Neural Networks (CNNs) have achieved excellent performance in agricultural image analysis studies [8, 9]. These algorithms can be used to accurately classify wheat varieties and identify subtle differences in wheat morphology that may be difficult to visually distinguish from wheat images alone by learning discriminative patterns from wheat image sets [17, 18]. Therefore, image processing along with machine learning techniques has emerged as a viable approach for wheat classification where efficiency, accuracy, repeatability, and scalability are enhanced.

Although significant strides have been made in the field of agricultural machine vision, current research has mainly concentrated on the detection of diseases, monitoring of crop health, and binary classification, compared to the relatively limited work on the classification of wheat varieties in multiple classes with real-field image data sets and the development of sophisticated feature extraction techniques [19, 20]. Moreover, the establishment of advanced classification schemes that can differentiate the very similar wheat cultivars for a variety of conditions is an open research problem. Wheat is highly significant crop in Pakistan and as the demand for intelligent agricultural decision support systems is increasing, it is highly scientific and significant to develop an accurate and automated wheat classification framework.

Hence, this work aims to explore the use of machine learning and computer vision technologies for wheat variety classification. By integrating digital image processing, feature extraction, and advanced classification algorithms, the proposed framework aims to improve the accuracy and reliability of wheat classification. The approach is intended to support wheat variety identification and may have potential applications in seed certification and agricultural decision support.

II. Literature Review

With the advent of machine learning, computer vision, and deep learning, agricultural research has undergone a transformation, especially in the field of wheat seed and variety classification. Classification of wheat is essential to ensure varietal purity, crop management, seed certification, and better agricultural productivity. Practically, farmers' wheat seed identification methods are largely based on manual work and the expertise of experts. These methods are also time-consuming, subjective, and challenging to scale up. For this reason, automated image-based classification systems have been used. The classification is achieved by the use of digital image processing and machine learning algorithms to enhance its accuracy, efficiency and uniformity.[21, 22].

Initially researchers used handcrafted visual features of the wheat seed images such as color, shape, texture, area, perimeter, and geometric features. Wheat classification and seed quality assessment were often conducted with traditional machine learning algorithms like Support Vector Machines (SVM), Decision Trees, Random Forests, Naïve Bayes, and Artificial Neural Networks. The performance of these approaches was found to be encouraging, with classification accuracy frequently exceeding 90% [22-25]. SVM consistently performed better than other traditional classifiers in high-dimensional feature spaces and with limited training samples [21, 26, 27].

Wheat image analysis has also been largely influenced by texture-based feature extraction techniques. Gray-Level Co-occurrence Matrix (GLCM) descriptors and their related texture measures are broadly used to identify the discriminative visual patterns in wheat seed images and plant images. Several studies have demonstrated that texture features, in combination with machine learning classifiers, play a significant role in improving the classification result and also offer additional texture information beyond color and geometric measurements [28, 29]. Likewise, morphological and morphometric traits like kernel length, width, degree of kernel compactness, and groove dimensions are highly informative for wheat variety identification [26, 30].

Despite the limitations of handcrafted feature engineering, deep learning has emerged as a rapidly advancing field, thus moving research from manual feature engineering to automatic feature learning. In wheat classification, Convolutional Neural Networks (CNNs) are gaining popularity as the most prominent approach due to their capacity to extract complex visual features directly from raw images. Several CNN models are able to recognize tremendous classification accuracy on wheat seed datasets. A custom CNN architecture attained an accuracy of more than 97% for wheat classification [31], whereas transfer learning approaches with pretrained networks, including DenseNet201, MobileNet, and InceptionV3, revealed their effectiveness in improving classification performance [32]. More recent studies on Advanced CNN-based architectures have attained higher accuracy, exceeding 98% for wheat variety recognition [33-35].

Recently, there are also efforts on low-power deep learning algorithms for mobile devices and edge devices. Agricultural research is usually conducted in resource-limited environments, and decreasing the computational complexity without

compromising accuracy has become an important research goal. LWheatNet attained 98.59% classification accuracy with a model size of almost 1.33 MB [34]. Similarly, the G-PPW-VGG11 model for wheat field-based classification supports that lightweight networks can be used for real-time agricultural applications in achieving high accuracy. The studies indicate a shift towards more application-focused approach and smart farming [20].

Attention mechanisms and transformer-based architectures, which allow the model to focus on the most informative visual regions, have also improved the classification performance. Hybrid network models of CNN, attention modules, and recurrent networks have exhibited the best performance for learning delicate distinctions between wheat varieties having similar appearances. As an example, A hybrid Vision Transformer-CNN model attains an accuracy of 98.33% [36], and Xception-BiLSTM model achieves an accuracy of 97.73% [37]. Hard-voting groups also show good results with classification accuracy of up to 99% [33].

Hyperspectral imaging has also been identified as a potential non-destructive technique for wheat classification and for assessing wheat seed quality along with the RGB imaging. Hyperspectral sensors work differently from the standard imaging system. They are able to provide both spatial and spectral data, which are necessary for detailed characterization of wheat varieties. Multiple studies have combined hyperspectral imaging with machine learning algorithms, including SVM, Partial Least Squares Discriminant Analysis (PLS-DA), XGBoost, deep neural networks, and others, to deliver excellent classification accuracy [38-42]. The hyperspectral approaches show good performance, but they are costly to acquire and computationally intensive.

Another relevant research line is in the field of image segmentation and object detection techniques, which could be useful in enhancing the extraction of features, such as the isolation of wheat grains, spikes, and heads from the background. To meet the particular demands of agricultural production, several contemporary segmentation architectures have emerged to significantly enhance wheat object detection and wheat segmentation in real-world agriculture, including DeepLabV3+ [43], CBAM-HRNet [44], YOLOv8-HD [45]and semi-supervised encoder-decoder networks [46]. These advances enable more reliable feature extraction, resulting in improved classification performance.

In Pakistan, the study on the analysis of wheat using AI has significantly increased over the past few years. Local research has been done on wheat diseases, growth stage identification, yield prediction, and dataset development. A benchmark dataset with standardized images of major wheat varieties was presented, which is crucial for future machine learning research in Pakistan [47]. A hybrid MobileNetV2-DenseNet121 network was designed to obtain a recognition rate of about 99% for wheat growth stages [48]. Deep learning and image processing methods have been effectively used in wheat disease detection and yield prediction in the field and through remote sensing [49-51].

Although there are enormous advances in wheat classification studies, there are still some critical questions that have not been answered. Most of the previous studies have used publicly available international datasets or laboratory settings, which may not apply to local agricultural conditions [22, 25, 26]. Furthermore, Pakistani local wheat variety image datasets have been used in a limited number of studies [47]. Although deep learning, transformer-based, and ensemble methods have shown high classification accuracies, many of these approaches require considerable computational resources. This reduces their suitability for deployment on mobile devices and in resource-constrained agricultural environments [20, 33, 36]. Additionally, the focus of most of the existing studies is only on classification performance, and little attention is paid to the translation of classification results into practical recommendations for agronomic operations that can be used directly by farmers in their decision-making processes.

Hence, robust and efficient machine learning models are required to achieve precise Pakistani wheat variety classification using image-based data. Such systems should also support practical agricultural decision-making by linking variety classifications with crop management. These can contribute to precision agriculture and sustainable wheat production.

III. METHODOLOGY

A. Proposed Methodology

The main goal of the study is to propose an automated method of identifying wheat seed varieties based on digital image data. The proposed framework is depicted in Figure 1 that describes the entire process of classifying wheat seed varieties. These six steps represent the entire process.

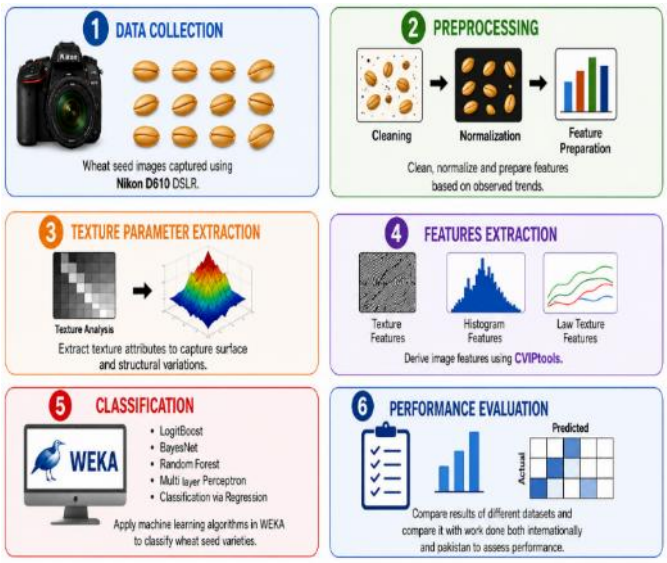


Fig. 1. Proposed methodology framework.

B. Study Area and Wheat Varieties

A total of six wheat varieties were used in the study, which are commonly grown in Southern Punjab, Pakistan, namely Akbar-19, Dilkash-20, Inqalab-91, Sadiq-21, Nawab-21 and Ghazi-19. Wheat seed samples were gathered from Bahawalpur, Multan, and Dera Ghazi Khan districts. These varieties were carefully chosen because of their agricultural importance and widespread cultivation in these areas.

C. Image Acquisition

Wheat seeds images were captured by using a Nikon D610 DSLR camera equipped with a 24-megapixel sensor under controlled studio lighting. Each of the 300 images was a single physical sample of seed, and 50 samples were taken from each of the six varieties: 50 x 6 = 300 images. So, no repeated images of the same physical seed were present throughout the dataset. Environmental variability was minimized by maintaining a fixed camera setup, a uniform background, and consistent lighting throughout the image acquisition process. Sample images of each variety used in this research are shown in Figure 2.

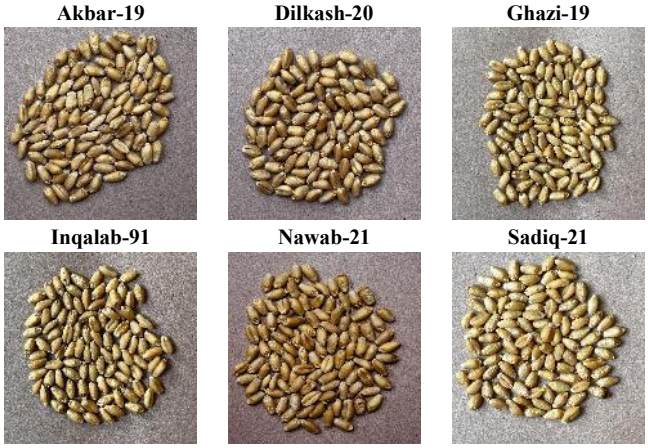


Fig. 2. Wheat seed images.

D. Image Preprocessing

The captured images were resized to three different resolutions: 256×256, 512×512, and 1024×1024 pixels to explore the effect of image scale on classification accuracy. RGB images were converted into grayscale images to simplify the computation required while maintaining structural and texture information. Image enhancement and normalization techniques were applied to improve contrast and reduce noise. Preprocessing enhanced image quality and normalized the dataset for analysis. Given an image $I(x, y)$, normalization is computed as:

$$I_n(x, y) = \frac{I(x,y)-\mu}{\sigma} \quad (1)$$

where μ is the mean intensity and σ denotes the standard deviation. This step helps to minimize illumination variations and improve relevant features. Noise reduction techniques were also applied to improve image clarity.

E. Feature Extraction

CVIPtools software was used for feature extraction. Three types of descriptors were extracted: histogram features, Gray-Level Co-occurrence Matrix (GLCM) texture features, and Laws texture features. Five regions of interest (ROIs) on individual seeds were selected randomly from each image to capture local texture variations, as tabulated below. Based on ROI circle size in Table 1, nine datasets comprising 91 features(Histogram, Texture, and Laws Texture) were extracted.

TABLE I. SUMMARY OF ROI USED FOR EACH DATASET

Sr. No	Datasets	ROI Circle Radius Size
	256 x 256	12 Pixels 14 Pixels 16 Pixels
	512 x 512	14 Pixels 16 Pixels 18 Pixels
	1024 x 1024	16 Pixels 18 Pixels 20 Pixels

F. Histogram Features

Histogram analysis represents the intensity distribution of image pixels [52]:

$$p(i) = \frac{n_i}{N} \quad (2)$$

where $\boldsymbol{n_i}$is the number of pixels with intensity $\boldsymbol{i}$and $\boldsymbol{N}$is the total number of pixels**.**

Mean:

$$\mu = \sum_{i=0}^{L-1} i \cdot p(i) \quad (3)$$

Variance**:**

$$\sigma^2 = \sum_{i=0}^{L-1} (i-\mu)^2 p(i) \quad (4)$$

Entropy**:**

$$H = -\sum_{i=0}^{L-1} p(i) \log p(i) \quad (5)$$

G. Gray-Level Co-occurrence Matrix (GLCM) Features

GLCM was employed to quantify spatial relationships between neighboring pixels. Texture features were computed in four orientations (0°, 45°, 90°, and 135°). The principal Haralick descriptors used in this study include Contrast, Correlation, Homogeneity, Energy (Angular Second Moment), and Entropy [53].

$$P(i,j) = \frac{\text{Frequency of pixel pair } (i,j)}{\text{Total number of pairs}} \quad (6)$$

Derived features include:

Contrast:

$$\text{Contrast} = \sum_{i,j} (i-j)^2 P(i,j) \quad (7)$$

Energy:

$$\text{Energy} = \sum_{i,j} P(i,j)^2 \quad (8)$$

Correlation:

$$\text{Correlation} = \frac{\sum_{i,j}(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j} \quad (9)$$

Homogeneity:

$$\text{Homogeneity} = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \quad (10)$$

These measures capture structural and spatial properties of wheat seed textures.

H. Laws Texture Features

Laws texture energy measures were extracted using convolution masks derived from the one-dimensional kernels L5, E5, S5, W5, and R5. These filters capture texture patterns associated with level, edge, spot, wave, and ripple structures. Symmetric masks were combined to reduce feature redundancy and improve computational efficiency.

- L, E, S, W, R represent:
  - L (Level): Average or smooth intensity in the image.
  - E (Edge): Edge detection (intensity changes) for texture.
  - S (Spot): Identifies spot-like patterns in the image.
  - W (Wave): Captures wave or oscillatory patterns.
  - R (Ripple): Identifies fine ripple or repetitive texture.

Law's texture energy measures use convolution masks to extract local texture information (Laws, 1980). The masks are formed using 1D vectors such as:

Level (L5): $[1\ 4\ 6\ 4\ 1]$

Edge (E5): $[-1\ -2\ 0\ 2\ 1]$

Spot (S5): $[-1\ 0\ 2\ 0\ -1]$

The 2D mask is generated as:

$$\text{Mask} = V_i^T \cdot V_j \tag{11}$$

Texture energy is computed as:

$$E = \frac{1}{MN} \sum_{x,y} | I(x,y) * \text{Mask} | \tag{12}$$

where $*$denotes convolution.

I. Feature Optimization

Principal Component Analysis (PCA) and Correlation-based Feature Selection (CFS) were used separately to reduce the dimensionality and remove redundant information from the extracted feature sets. PCA was used to reduce the size of the feature space while retaining the major variance in the data, whereas CFS was used to select attributes with high relevance and low inter-correlation. For each image resolution, PCA and CFS were applied separately to the complete extracted feature dataset, and separate data files were generated for the PCA-optimized and CFS-optimized feature sets. The resulting datasets contained 25 PCA features and 21 CFS-selected features, respectively. These optimized datasets were subsequently used for classification with the selected machine learning algorithms in WEKA. The classification performance was evaluated using 10-fold cross-validation. Thus, PCA and CFS were performed before the cross-validation stage and were not independently fitted within each training fold.

In general, the extracted feature vector is:

$$X_{opt} = [f_1, f_2, \dots, f_m] \tag{13}$$

where:

X_{opt}= optimized feature vector after feature reduction.

$f_1, f_2, \dots, f_m$= selected features.

m= number of retained features after optimization.

J. Classification Models

Five supervised machine learning algorithms implemented in WEKA 3.8.2 were evaluated: LogitBoost, BayesNet, Classification via Regression, Multi-Layer Perceptron (MLP), and Random Forest using default settings. In order to ensure uniformity, the same configurations were applied to all feature-optimized datasets for performance comparison. The classification performance was evaluated using 10-fold cross-validation. The selected algorithms have been used in agricultural classification tasks. These models learn a mapping function:

$$f: \mathrm{X} \rightarrow Y \tag{14}$$

where Y denotes the class label of wheat seed varieties.

K. Performance Evaluation

The 10-fold cross-validation was performed on the PCA-optimized and CFS-optimized datasets after the respective feature-selection procedures had been completed. The classification performance was measured by the following performance metrics: Accuracy, Precision, Recall, and F1-score. The classifiers were compared based on these performance measures to determine the most effective classifier. The classification results were also compared with recent studies from Pakistan and other countries that have been reported in the literature. The evaluation of the performance of the models is performed with classification measures:

Accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

Precision:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{16}$$

Recall:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{17}$$

F1-Score:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

A confusion matrix is used to visualize classification performance. Comparative evaluation is carried out among classifiers and benchmarked with existing national and international studies.

IV. Results

Six Pakistani wheat varieties were considered: Akbar-19, Dilkash-20, Ghazi-19, Inqalab-91, Nawab-21 and Sadiq-21. Digital images of these varieties were taken and used to evaluate the proposed wheat seed classification system. The experiments were carried out using three image resolutions (256×256, 512×512, and 1024×1024 pixels) to explore the effect of image quality on classification performance. GLCM having Texture features, Histogram features, and LBP texture features were obtained for each ROI in wheat seed regions.

Principal Component Analysis (PCA) and Correlation-based Feature Selection (CFS) were used for dimensionality reduction and feature selection, respectively. PCA was performed in WEKA using a cumulative variance retention threshold of 95% (-R 0.95), resulting in 25 principal components. CFS was performed using CfsSubsetEval with BestFirst search in the forward direction and default search parameters, resulting in 21 selected features. CfsSubsetEval evaluates feature subsets by considering the correlation between the selected features and the class while reducing redundancy among the selected features. The resulting PCA and CFS feature sets were used separately for classification at each image resolution.

The optimized feature sets were then classified using different machine learning algorithms. Previous research in this field has also reported that image-based feature extraction and machine learning provide an effective approach for automated wheat variety identification [24, 54].

The classification accuracy varied with the dimensionality reduction and feature selection methods. The major variance patterns were preserved, and important discriminative texture information was retained by PCA, giving better results than CFS. While CFS reduced the number of features, some information related to the seed surface characteristics was lost. The results showed that there are some subtle differences in the texture of the different wheat varieties and that the ability to preserve these differences is essential for the correct classification.

The performance of the machine learning classifiers varied with the selected features and image resolution in the 10-fold cross-validation experiments. Higher-resolution images generally delivered higher classification accuracy. This may be because they kept more detailed information about seed texture, shape, and surface patterns.

A. Classification Performance Analysis Using PCA Optimized Features

Table 2 presents classification performance of five machine learning classifiers using PCA-optimized features extracted from wheat seed images at three different resolutions (256×256, 512×512, 1024×1024).

TABLE II. ACCURACY COMPARISON OF CLASSIFIERS (PCA OPTIMIZED)

Classifier	Accuracy (%) Dataset 256 x 256	Accuracy (%) Dataset 512 x 512	Accuracy (%) Dataset 1024 x 1024
LogitBoost	86.00	91.73	92.40
BayesNet	84.07	92.13	88.80
ClassificatoinViaRegression	88.53	92.67	94.20
MultilayerPerceptron	87.60	91.53	93.80
RandomForest	92.80	94.67	96.40

The performance of five classification machine learning algorithms is presented for the three different resolution images of wheat seeds extracted from the images using PCA in Table 2. The corresponding classification accuracies are shown in Figure 3. The results reveal that the image resolution has a positive effect on the classification accuracy since the higher the image resolution, the more discriminative texture and structural information of the wheat seed variety could be obtained. The results also reveal that feature representation and the selection of the corresponding learning algorithm are crucial for classification accuracy.

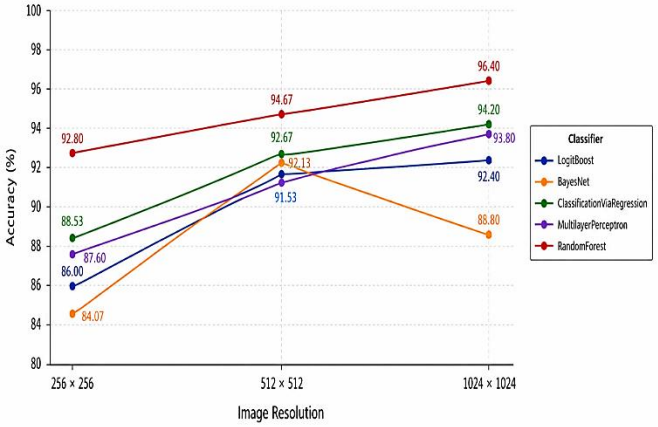


Fig. 3. Classification accuracy of different classifiers (PCA optimized).

Random Forest amongst all classifiers, has shown the highest accuracy for all image resolutions, with accuracies of 92.80%, 94.67%, and 96.40% for 256×256, 512×512, and 1024×1024 datasets, respectively. Random Forest outperforms other methods because it follows the ensemble learning method, with multiple decision trees working together to enhance the accuracy of classification and minimize errors.

The performance of ClassificationViaRegression was also found to steadily improve with increasing resolution of the images, with the results of 88.53%, 92.67%, and 94.20%, respectively. Likewise, LogitBoost and MultilayerPerceptron showed higher accuracy with higher-resolution images, suggesting that the higher-resolution images provided better feature representation. BayesNet performed well at 512×512 resolution but showed lower accuracy at 1024×1024 resolution, indicating a possible limitation in the ability of the model to capture higher dimensional relationships between the image feature dimensions.

The overall results indicate that PCA preserved important discriminative features in the wheat seed images. The combination of PCA and Random Forest achieved the best classification performance, and it is observed that it achieves the highest accuracy in the highest resolution data (96.40%). The results presented here emphasize the necessity of using image quality, feature optimization, and classifier selection in developing reliable wheat seed classification systems.

B. Classification Performance Analysis Using CFS Optimized Features

Table 3 shows the classification performance of five machine learning classifiers, trained with features optimized by CFS on the wheat seed images with three different image resolutions: 256×256, 512×512, and 1024×1024 pixels.

The results shown in Table 3 indicate that both the selected feature subset and the learning algorithm had influenced the classifier performance. CFS-based feature selection resulted in

lower classification accuracy as compared to PCA optimization. Figure 4 shows a graphical representation of their classification accuracy results. This indicates that some important discriminative information essentials for wheat variety classification was lost during feature reduction.

TABLE III. ACCURACY COMPARISON OF CLASSIFIERS (CFS OPTIMIZED)

Classifier	Accuracy (%) Dataset 256 x 256	Accuracy (%) Dataset 512 x 512	Accuracy (%) Dataset 1024 x 1024
LogitBoost	70.53	73.87	71.53
BayesNet	67.67	67.53	68.80
ClassificatoinViaRegression	79.07	77.33	**75.80**
MultilayerPerceptron	65.47	68.93	73.60
RandomForest	**85.27**	**84.33**	75.47

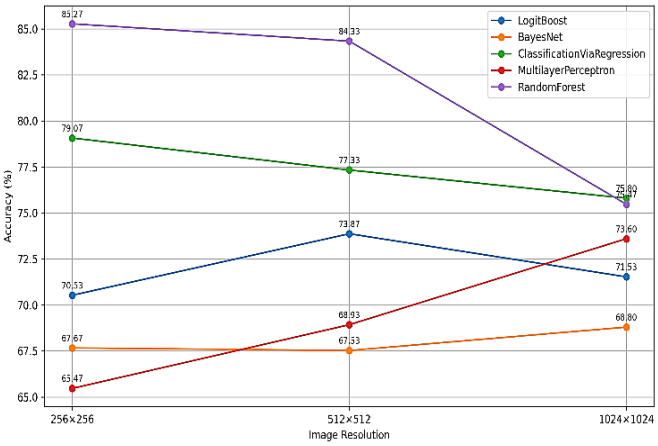


Fig. 4. Classification accuracy of different classifiers (CFS optimized).

In comparison, LogitBoost performed relatively stably on the image resolutions with an accuracy of 70.53%, 73.87% and 71.53% for 256×256, 512×512 and 1024×1024, respectively. In addition, BayesNet delivered similar results with a range of 67.53%-68.80%, indicating some consistency but not as effective as the other classifier in modelling complex relationships among the selected seed features.

ClassificationViaRegression achieved accuracies of 79.07%, 77.33%, and 75.80% for all the three datasets. However, its accuracy decreased slightly with increasing image resolution. This suggests that the CFS-selected features did not fully keep the additional information available in higher-resolution images. In contrast, MultilayerPerceptron showed gradual improvement from 65.47% to 73.60%, indicating better utilization of additional feature information at higher resolutions.

Random Forest achieved the highest accuracy for the 256×256 and 512×512 datasets, with accuracies of 85.27% and 84.33%, respectively. However, its performance decreased to 75.47% for the 1024×1024 dataset. This reduction suggests that the CFS-selected feature subset was less effective in representing complex patterns from higher-resolution images. Overall, the results confirm that although CFS reduced the feature dimensionality, PCA optimization retained more informative features and provided superior classification performance for wheat seed variety classification.

C. Performance Comparison of Both PCA and CFS Optimized Features

Table 4 and Table 5 report the performance of the proposed model and baseline methods with various evaluation measures. These metrics provide complementary information for overall classification accuracy, class-wise prediction performance, and discriminative ability. This allows a more comprehensive comparison of model effectiveness.

TABLE IV. ACCURACY COMPARISON OF CLASSIFIERS (PCA OPTIMIZED)

Classifier	Resolution	Kappa Statistic	Weighted F-Measure	ROC Area
LogitBoost	256×256	0.8320	0.860	0.982
LogitBoost	512×512	0.9008	0.917	0.995
LogitBoost	1024×1024	0.9088	0.924	0.994
BayesNet	256×256	0.8088	0.841	0.980
BayesNet	512×512	0.9056	0.921	0.996
BayesNet	1024×1024	0.8656	0.888	0.991
ClassificatoinVia Regression	256×256	0.8624	0.885	0.985
ClassificatoinVia Regression	512×512	0.9120	0.932	0.995
ClassificatoinVia Regression	1024×1024	0.9304	0.968	0.992
MultilayerPercept ron	256×256	0.8512	0.876	0.976
MultilayerPercept ron	512×512	0.8984	0.915	0.990
MultilayerPercept ron	1024×1024	0.9256	0.938	0.992
RandomForest	256×256	0.9136	0.928	0.994
RandomForest	512×512	0.9360	0.947	0.997
RandomForest	**1024×1024**	**0.9568**	**0.964**	**0.998**

TABLE V. ACCURACY COMPARISON OF CLASSIFIERS (CFS OPTIMIZED)

Classifier	Resolution	Kappa Statistic	Weighted F-Measure	ROC Area
LogitBoost	256×256	0.6464	0.703	0.943
LogitBoost	512×512	0.6864	0.739	0.952
LogitBoost	1024×1024	0.6584	0.708	0.942
BayesNet	256×256	0.612	0.675	0.933
BayesNet	512×512	0.6104	0.677	0.938
BayesNet	1024×1024	0.6256	0.68	0.926
Classificatoin ViaRegressio n	256×256	0.7488	0.789	0.965
Classificatoin ViaRegressio n	512×512	0.728	0.773	0.963
Classificatoin ViaRegressio n	1024×1024	0.7096	0.747	0.954
MultilayerPer ceptron	256×256	0.5856	0.655	0.909
MultilayerPer ceptron	512×512	0.6272	0.691	0.918

MultilayerPerceptron	1024×1024	0.6832	0.728	0.942
RandomForest	**256×256**	**0.8232**	**0.852**	**0.974**
RandomForest	512×512	0.812	0.843	**0.977**
RandomForest	**1**024×1024	0.7056	0.753	0.949

The Random Forest classification rate was 96.4% but was also supplemented with other evaluation parameters such as Precision, Recall, F1-Score, Specificity and AUC-ROC presented in Tables 4 and 5. These metrics provide alternative information about the accuracy of the classifier in correctly predicting positive and negative cases while considering false-positive and false-negative rates and in properly separating classes. The overall high values for all these metrics justify the classification rate reported as 96.4% for the Random Forest classifier and give evidence of good and reliable classification.

D. Comparison with International Research

The results obtained were compared with previous international wheat classification studies. Kayabaşı et al. [54] achieved approximately 93% accuracy using texture and morphological features with artificial neural networks. Basati et al. [24] reported around 90.20% accuracy using color, morphology, and texture features with decision tree classification. Singh et al. [55] reported 97.70% accuracy using advanced feature representation and fuzzy clustered random forest. Lüy et al. [28] achieved 82.20% accuracy by using GLCM texture features and grain characteristics as mentioned in Table 6.

TABLE VI. COMPARISON WITH INTERNATIONAL STUDIES ON WHEAT SEED CLASSIFICATION USING TEXTURE/IMAGE FEATURES AND MACHINE LEARNING

Study	Features Used	Classifier	Accuracy (%)	Comparison with Present Study
Kayabaşı et al. (2017)	Image texture and morphological features	ANN	93.00	Present study achieved higher accuracy (96.40%) using PCA and RandomForest
Basati et al. (2018)	Color, morphology, and texture features	Decision Tree (J48)	90.20	The present study improved classification accuracy by more than 6%.
Singh et al. (2020)	Visual grain characteristics	Fuzzy Clustered Random Forest	97.70	Slightly higher than the present study; however, it focused on grain grading rather than classification of multiple wheat varieties.
Lüy et al. (2023)	GLCM texture features and grain characteristics	Machine Learning	82.00	The present study substantially outperformed the reported results that time.
Present Study	GLCM texture and intensity features + PCA	RandomForest	**96.40**	The present study achieved 96.40% accuracy among reviewed texture-based varietal classification studies.

Studies are noted in Table 6, with the difference among them in respect to the number of wheat classes, the number of datasets, the setup of acquiring images, image preprocessing, and the way the studies are validated. All of this can have significant impact on classification accuracy and on the validity of direct one-on-one comparisons. The reported accuracies are therefore listed, in particular, to provide a benchmark in this research. Compared with these studies, the proposed PCA-based Random Forest framework achieved competitive performance while using a comparatively simple and computationally efficient machine learning approach.

E. Comparison with Pakistani Research Studies

The proposed method was also compared with recent studies conducted in Pakistan, as shown in Table 7.

TABLE VII. COMPARISON WITH RESEARCH CONDUCTED IN PAKISTAN

Study	Research Focus	Method	Accuracy (%)	Comparison with Present Study
Nawaz et al. (2025)	Development of Pakistani wheat seed dataset	Dataset Creation	N/A	Provides a dataset but not a classification framework.
Yasar et al. (2024)	Bread wheat variety classification	CNN + SVM	94.00	GLCM features were not used.
Yasar et al. (2024)	Bread wheat variety classification	ResNet18s	97.57	GLCM features were not used.
Usman et al. (2024)	Wheat seed disease classification	CNN	98.43	Different tasks focusing on disease detection rather than variety classification. Further, CNN-based methods are more accurate, but they normally require more data, GPU resources, and long training time. A quantitative comparison of computational resources was not performed in the present study.
Ashraf et al. (2023)	Wheat disease recognition	CNN	93.00	Different application domain.
Present Study	Six Pakistani wheat varieties	PCA + RandomForest	96.40	One of the highest accuracies reported using traditional machine-learning techniques.

The performance of the proposed wheat seed classification framework was compared with recent studies conducted in Pakistan, as summarized in Table 7. Existing studies have mainly focused on wheat variety classification using deep learning approaches or wheat disease recognition, whereas the present study focuses on automatic classification of six Pakistani wheat varieties using texture-based features and traditional machine learning. Further, the accuracy reported is a contextual benchmark for wheat seed classification research in Pakistan. But caution should be taken in making direct one-to-one comparisons due to the differences in the size of the data sets, wheat varieties, the way the images were taken, the preprocessing methods, and how the images were validated. These factors may influence the accuracy in the classification and can have an impact on the comparability of accuracy values reported.

Nawaz et al. [47] developed a Pakistani wheat seed image dataset, which provides an important resource for future research; however, the study did not present a complete classification framework. Yasar et al. [56] achieved 94.00% and 97.57% accuracy using CNN-SVM and ResNet18 models, respectively, for bread wheat variety classification. However, these approaches relied on deep learning architectures and did not utilize GLCM-based handcrafted texture features. Similarly, Usman et al. [51] and Ashraf et al. [49] reported high accuracies using CNN models, but their studies focused on wheat disease detection rather than variety identification.

In comparison, the proposed PCA-optimized Random Forest framework attains an accuracy of 96.40% for six Pakistani wheat varieties. The results show that the proposed approach presents competitive performance with deep learning-based approaches, requiring comparatively lower computational complexity using handcrafted features and a Random Forest classifier. A quantitative comparison of computational complexity with deep learning approaches was not performed in this study. The results highlight the potential of optimized texture features along with machine learning algorithms for wheat seed variety classification.

F. Research Contributions

This work provides a framework for wheat seed variety classification using image acquisition, image preprocessing, texture and intensity-based feature extraction, feature dimensionality reduction and selection, and machine learning. The framework can support research in agricultural image analysis and wheat seed classification. The framework proposed is applicable to small and medium-sized datasets in the agricultural context and can be adapted to other classification problems of crops. The comparison of five classifiers (LogitBoost, BayesNet, ClassificationViaRegression, MultilayerPerceptron, and RandomForest) shows that the choice of a particular classifier, as well as feature optimization, has a pronounced effect on the quality of the classification result. The wheat variety identification results demonstrated that the PCA-optimized RandomForest model gave the highest accuracy of 96.40%; thus, it is effective in wheat variety identification tasks. This study offers fresh empirical data on the texture features of wheat varieties grown in South Punjab, Pakistan, and fills the gap of wheat seed image datasets and classification studies in the region. The proposed method shows that the handcrafted texture features and the traditional machine learning algorithms can accurately distinguish wheat seed varieties without being destructive, quickly, and cheaply. The proposed framework has potential applications in seed identification, varietal verification, and agricultural image analysis. It may also provide a basis for future research involving deep learning, hybrid feature extraction, and larger agricultural datasets.

V. CONCLUSION

In this study, a complete image-based wheat variety classification framework based on machine learning techniques was presented. Whereas the previous work was performed on benchmark datasets like the UCI Wheat Seeds dataset, this study is based on a real-world image dataset of six wheat varieties from Pakistan, namely Akbar-19, Dilkash-20, Ghazi-19, Inqalab-91, Nawab-21, and Sadiq-21. The use of locally cultivated wheat varieties with similar visual characteristics makes the classification problem more difficult and practically relevant. The study provides a detailed image-based classification framework for Pakistani wheat varieties. It also addresses the limited availability of local wheat seed image datasets.

Five machine learning classifiers, including LogitBoost, BayesNet, ClassificationViaRegression, MultilayerPerceptron, and RandomForest, were systematically compared in their performance, with identical experimental conditions. The results showed that the choice of classifiers has a remarkable impact on the classification accuracy, with the highest accuracy of 96.40% achieved by PCA-optimized features with RandomForest. This study also examined the influence of the image resolution, by comparing the performance of image datasets with resolutions of 256×256, 512×512 and 1024×1024, with higher resolution images generally yielding better classification results as more discriminative texture and intensity information was available.

One of the key findings of this research is that handcrafted features used for texture and intensity are employed instead of deep learning-based features extraction. The results show that traditional machine learning methods can achieve high classification accuracy using handcrafted texture and intensity features. The present study did not quantitatively evaluate computational requirements or compare them with deep learning models. Therefore, computational efficiency should be investigated separately in future work. In addition, the comparison of PCA and CFS feature optimization methods demonstrated that PCA preserved more informative features and that it gave better classification results when identifying wheat seeds.

VI. LIMITATIONS

Overall, the proposed framework was able to provide good classification performance, but there are some limitations that should be considered. The dataset used includes six varieties of wheat from Pakistan and was sampled under controlled imaging conditions; thus, the model needs to be tested with more varieties, different regions, and different environmental conditions. Second, the image acquisition process was conducted under relatively controlled imaging conditions, which may induce some variations in illumination, background

noise, seed orientation, and image quality in an actual agricultural environment that can impact the classification performance. Third, the study is primarily based on handcrafted texture and intensity features, which can achieve good classification results, but the performance of the proposed handcrafted-feature approach was not directly compared with deep learning-based feature extraction in this study. In addition, the data size is relatively small, especially when compared to the large-scale deep learning applications. Future work needs to be directed towards building larger and more diverse wheat seed image datasets and testing the hybrid or deep learning approach to gain robustness and scalability.

VII. Future Work

The proposed framework provides a trustworthy, inexpensive, and efficient solution for wheat variety classification. This performance achieved shows its possible use in seed certification, verification of variety, grain quality, and smart systems in agriculture. The dataset developed, methodology, class performance evaluation, and experimental results will serve as a benchmark for future research on deep learning-based frameworks, hybrid feature extractors, seed quality analysis, and large-scale agricultural automation in the context of agriculture in Pakistan and other developing countries.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Information Technology, The Islamia University of Bahawalpur, for providing the facilities and support required to complete this research. The authors also thank all individuals who assisted in the collection and preparation of the wheat seed dataset. Their support and valuable contributions are greatly appreciated.

References

[1] ADDIN EN.REFLIST Food and N. Agriculture Organization of the United Nations, *World Food and Agriculture - Statistical Yearbook 2020*. 2020: Food and Agriculture Organization of the United Nations.

[2] Stella, T., et al., Wheat crop traits conferring high yield potential may also improve yield stability under climate change. in silico Plants, 2023. 5(2): p. diad013.

[3] Shahzad, A., et al., *Current Situation and Future Prospects of Wheat Production in Pakistan.* Journal of Life and Social Sciences, 2022. 2022(1): p. 1–12.

[4] Ahmed, M., A. Sameen, and A.M.S. Kheir, Warming trends and shortened growing seasons: integrating four decades of observations and model simulations to develop wheat adaptation stratsegies in semi-arid Pakistan. Sci Rep, 2026. 16(1): p. 4766.

[5] Shahid, M.A., et al., Assessment of Water Productivity Enhancement and Sustainability Potential of Different Resource Conservation Technologies: A Review in the Context of Pakistan. Agriculture, 2022. 12(7): p. 1058.

[6] Ahmad, J., et al., 'Mh-21,' a Novel High-Yielding and Rusts Resistant Bread Wheat Variety for Irrigated Areas of Punjab, Pakistan. SABRAO Journal of Breeding and Genetics, 2023. 55(3): p. 749-759.

[7] Bhutta, S.K., et al., Evaluation of Growth and Yield Attributes of Some Wheat Varieties under Local Conditions of Southern Punjab, Pakistan. Journal of Plant Breeding and Genetics, 2019. 7(1): p. 1-7.

[8] Ceyhan, M., et al., *Classification of wheat varieties with image-based deep learning.* Multimedia Tools and Applications, 2023. 83(4): p. 9597-9619.

[9] Gao, J., et al., Identification Method of Wheat Cultivars by Using a Convolutional Neural Network Combined with Images of Multiple Growth Periods of Wheat. Symmetry, 2021. 13(11): p. 2012.

[10] Ghazal, S., A. Munir, and W.S. Qureshi, Computer Vision in Smart Agriculture and Precision Farming: Techniques and Applications, in Artificial Intelligence in Agriculture (Review). 2024.

[11] Hamrani, A., et al., AI and Robotics in Agriculture: A Systematic and Quantitative Review of Research Trends (2015–2025). Crops, 2025. 5(5): p. 75.

[12] Upadhyay, A., et al., Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. Artificial Intelligence Review, 2025. 58(3): p. 92.

[13] Yu, F., et al., Advancements in Utilizing Image-Analysis Technology for Crop-Yield Estimation. Remote Sensing, 2024. 16(6): p. 1003.

[14] Wang, et al., Estimation of agronomic characters of wheat based on variable selection and machine learning algorithms. Agronomy, 2023. 13(11): p. 2808.

[15] Shin, J., et al., Trends and Prospect of Machine Vision Technology for Stresses and Diseases Detection in Precision Agriculture. AgriEngineering, 2022. 5(1): p. 20-39.

[16] Pablo Guerra, J. and F. Cuevas, Application of Digital Image Processing Techniques for Agriculture: A Review, in Digital Image Processing - Latest Advances and Applications. 2024, IntechOpen.

[17] Ahmed, N., et al., Enhancing food security: machine learning-based wheat yield prediction using remote sensing and climate data in Pakistan. Theoretical and Applied Climatology, 2025. 156(5).

[18] Machine, V. and L. Intelligent Systems, Multi-year Wheat Mapping from Satellite Time Series Leveraging Superpixel-Based Active Learning for Data Sparse Regions. 2023.

[19] Khan, H., et al., Automated Wheat Diseases Classification Framework Using Advanced Machine Learning Technique. Agriculture, 2022. 12(8): p. 1226.

[20] Pan, Y., et al., Classification of field wheat varieties based on a lightweight G-PPW-VGG11 model. Front Plant Sci, 2024. 15: p. 1375245.

[21] Fazel, N., et al., Quality assessment of components of wheat seed using different classifications models. Applied Sciences, 2022. 12(9): p. 4133.

[22] Khatri, et al., Wheat seed classification: utilizing ensemble machine learning approach. Scientific programming, 2022. 2022(1): p. 2626868.

[23] Baryshev, et al., Comparison of machine learning methods for solving the problem of wheat seeds classification by yield properties. Russian Agricultural Sciences, 2020. 46(4): p. 410-417.

[24] Basati, et al., Using different classification models in wheat grading utilizing visual features. International Agrophysics, 2018. 32(2): p. 225-235.

[25] 2Priya and G. B, *A comparison of various machine learning algorithms for wheat seed data set classification.* International Journal of Research in Advent Technology, 2019(Special): p. 39-43.

[26] Sun, et al., Research on Wheat Seed Classification Based on Machine Learning Algorithms and Data Analysis Visualization. Advances in Computer, Signals and Systems, 2025. 9(2).

[27] Tyagi, N., B. Raman, and N.M. Garg, Varietal Classification of Wheat Seeds Using Hyperspectral Imaging Technique and Machine Learning Models, in Computer Vision and Image Processing, D. Gupta, et al., Editors. 2023, Springer Nature Switzerland: Cham. p. 253-266.

[28] Lüy, M., et al., Investigation of the effect of hectoliter and thousand grain weight on variety identification in wheat using deep learning method. Journal of Stored Products Research, 2023. 102: p. Article.

[29] Shafi, U., et al., Wheat Yellow Rust Disease Infection Type Classification Using Texture Features. Sensors (Basel), 2021. 22(1): p. 146.

[30] Madhavan, et al., *Wheat seed classification using neural network pattern recognizer.* Materials Today: Proceedings, 2023. 81: p. 341-345.

[31] Lingwal, et al., *Image-based wheat grain classification using convolutional neural network.* Multimedia tools and applications, 2021. 80(28): p. 35441-35465.

[32] Karim, L., et al., *Wheat varieties identification based on a deep learning approach.* Journal of the Saudi Society of Agricultural Sciences, 2021. 20(5): p. 281-289.

[33] Bentaieb, S., A.Y. Boumedine, and A. Ouamri, *Enhancing wheat seed classification with ensemble voting of multiple fine-tuned pre-trained models.* Multimedia Tools and Applications, 2025. 84(38): p. 47261-47288.

[34] Guo, X., et al., LWheatNet: a lightweight convolutional neural network with mixed attention mechanism for wheat seed classification. Front Plant Sci, 2024. 15: p. 1509656.

[35] Xing, et al., Wheat-seed variety recognition based on the GC_DRNet model. Agriculture, 2023. 13(11): p. 2056.

[36] Dönmez, Hybrid convolutional neural network and multilayer perceptron vision transformer model for wheat species classification task: E-ResMLP+. European Food Research and Technology, 2024. 250(5): p. 1379-1388.

[37] Yasar, et al., Classification of bread wheat varieties with a combination of deep learning approach. European Food Research and Technology, 2024. 250(1): p. 181-189.

[38] Bao, et al., Rapid classification of wheat grain varieties using hyperspectral imaging and chemometrics. Applied Sciences, 2019. 9(19): p. 4119.

[39] Jin, et al., Spatial-spectral feature extraction of hyperspectral images for wheat seed identification. Computers and Electrical Engineering, 2022. 101: p. 108077.

[40] Li, Y., et al., Classification of Nitrogen-Efficient Wheat Varieties Based on UAV Hyperspectral Remote Sensing. Plants (Basel), 2025. 14(13): p. 1908.

[41] Zhang, H., et al., Detection of seed purity of hybrid wheat using reflectance and transmittance hyperspectral imaging technology. Front Plant Sci, 2022. 13: p. 1015891.

[42] Zhou, L., et al., Wheat Kernel Variety Identification Based on a Large Near-Infrared Spectral Dataset and a Novel Deep Learning-Based Feature Selection Method. Front Plant Sci, 2020. 11: p. 575810.

[43] Xu, X., et al., Segmentation and counting of wheat spike grains based on deep learning and textural feature. Plant Methods, 2023. 19(1): p. 77.

[44] Zhang, et al., A lightweight winter wheat planting area extraction model based on improved DeepLabv3+ and CBAM. Remote Sensing, 2023. 15(17): p. 4156.

[45] Ma, N., et al., Wheat Seed Detection and Counting Method Based on Improved YOLOv8 Model. Sensors (Basel), 2024. 24(5): p. 1654.

[46] Ghanbari, A., G.H. Shirdel, and F. Maleki, Semi-Self-Supervised Domain Adaptation: Developing Deep Learning Models with Limited Annotated Data for Wheat Head Segmentation. Algorithms, 2024. 17(6): p. 267.

[47] Nawaz, M., et al., High-resolution RGB image dataset for wheat seed varietal identification and purity assessment. Data Brief, 2025. 61: p. 111690.

[48] Naseer, A., et al., Novel hybrid transfer neural network for wheat crop growth stages recognition using field images. Sci Rep, 2025. 15(1): p. 11822.

[49] Ashraf, et al., *A Convolutional Neural Network Model for Wheat Crop Disease Prediction.* Computers, Materials & Continua, 2023. 75(2): p. 3867–3882.

[50] Rashid, et al., *Multispectral Approach for Wheat Yield Estimation Using Deep Learning.* Journal of Computing and Biomedical Informatic, 2025. 8(2).

[51] Usman, I., A. Wakeel, and S. Syed Muhammad Adnan, *Leveraging CIELAB Segmentation and CNN for Wheat Fungi Disease Classification.* International Journal of Innovations in Science & Technology, 2024. 6: p. 1045–1057.

[52] Gonzalez, R.C. and R.E. Woods, *Digital Image Processing*. 4th ed. 2018: Pearson.

[53] Haralick, R.M., K. Shanmugam, and I.H. Dinstein, *Textural Features for Image Classification.* IEEE Transactions on Systems, Man, and Cybernetics, 1973. SMC-3(6): p. 610-621.

[54] Kayabaşı, et al., Image processing based ann with Bayesian regularization learning algorithm for classification of wheat grains, in 10th International Conference on Electrical and Electronics Engineering (ELECO) 2017. p. 1166-1170.

[55] Singh, et al., Classification of wheat seeds using image processing and fuzzy clustered random forest. International Journal of Agricultural Resources, Governance and Ecology, 2020. 16(2): p. 123-156.

[56] Yasar, Analysis of selected deep features with CNN-SVM-based for bread wheat seed classification. European Food Research and Technology, 2024. 250(6): p. 1551-1561.

Perceptual Evaluation of Garment Drape Intensity in AI-Generated Garment Images

A Pilot Study Using a Semi-Automated Evaluation Pipeline with CLIP

Noriaki Kuwahara[1]*, Yuta Yamamoto[2], Dongeun Choi[3], Kazunari Hirakoso[4], Takashi Satou[5]
Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan[1, 2, 3]
Tamagawa University, 6-1-1 Tamagawa-Gakuen, Machida, Tokyo 194-8610, Japan[4]
Tokyo Denki University, 5 Senju-Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan[5]

***Abstract*—Recent advances in generative artificial intelligence have enabled fine-grained manipulation of visual attributes in garment images, but consistency in automated feature space does not necessarily guarantee consistency in human perception. This study investigates the perceptual validity of garment drape intensity control in AI-generated garment images using a semi-automated Human-in-the-Loop evaluation pipeline. Five base garments were manipulated across five latent-space control levels and evaluated independently using Contrastive Language–Image Pre-training (CLIP)-based feature-space analysis and a two-alternative forced-choice (2AFC) experiment with 15 participants. The CLIP projection scores increased monotonically with the manipulation parameter for all five garment sequences. Human judgments showed a similar monotonic tendency at the aggregate level; however, garment-specific Bradley–Terry analysis revealed local reversals in the estimated perceptual ordering for three of the five base garments. These results demonstrate that aggregate agreement between automated and human evaluation can mask stimulus-specific perceptual discrepancies. The proposed framework provides a practical approach for combining automated screening with human perceptual validation when assessing fine-grained attributes in generative fashion systems.**

***Keywords*—Drape perception; generative AI; human-in-the-loop; CLIP; bradley–terry model; fashion image generation**

I. Introduction

Generative models have increasingly enabled fine-grained control of visual attributes in synthesized images. Diffusion-based and latent diffusion models have provided a flexible foundation for high-quality and controllable image generation [1], [2]. In the fashion domain, generative models are being applied to garment generation, editing, virtual try-on, and design support [3]. Beyond generating visually plausible garments, an important challenge is to determine whether controlled changes introduced by generative models correspond to visual differences that are consistently perceived by human observers.

This issue is particularly relevant for perceptual garment attributes. A generative model may produce systematic changes in its latent representation or in an automated feature space, but such changes do not necessarily imply equally systematic changes in human perception. Recent work on human-preference-based evaluation of generated images has also highlighted the importance of evaluating correspondence between automated representations and human judgments [4]. Drape provides a useful case for examining this problem because its visual impression emerges from multiple garment-dependent characteristics, including folds, silhouette, and the apparent manner in which fabric hangs. The perceived strength of drape may therefore depend not only on the magnitude of the generative manipulation but also on the visual context of the individual garment.

Our previous study proposed an evaluation framework combining latent-space analysis, CLIP-based automated evaluation, and human perceptual judgments to assess the reliability of attribute control in diffusion-based fashion generation [5]. The present study extends this approach to garment drape as a textile-oriented perceptual attribute. Latent-space manipulation is employed because it enables systematic variation of a target attribute along a continuous control parameter while using the same base garment as a reference. This controlled setting is particularly useful for perceptual evaluation, where changes in the target attribute should be distinguished from unrelated variations in garment appearance. Rather than examining only whether an automated score changes monotonically with the manipulation parameter, we investigate whether the corresponding perceptual ordering is reproduced by human observers and whether this correspondence remains consistent across individual base garments. In this study, the term drape intensity is used operationally to denote the perceived strength of drape in AI-generated garment images and is not intended to represent a conventional physical property of textiles.

To address these questions, we employ a semi-automated Human-in-the-Loop evaluation pipeline integrating latent-space quality control, CLIP-based feature-space evaluation, and a two-alternative forced-choice (2AFC) perceptual experiment analyzed using the Bradley–Terry model. The main contributions of this study are threefold: 1) we operationalize drape intensity as a perceptual attribute for controlled evaluation of AI-generated garment images; 2) we apply and extend our previously proposed semi-automated evaluation framework to garment drape, with automated feature-space behavior and human perceptual judgments evaluated independently and subsequently compared; and 3) we demonstrate that aggregate agreement between automated and human evaluation can mask garment-specific perceptual discrepancies. Through this analysis, the study examines the role of human perceptual validation in assessing fine-grained attribute control in generative fashion systems.

*Corresponding author.

II. Related Work

A. Textile Drape Measurement and Perceptual Evaluation

Drape has long been studied in textile and apparel engineering as a property arising from the interaction between fabric mechanical characteristics, gravity, and garment construction. Early quantitative approaches introduced drape coefficients and related geometric measures to characterize the deformation of fabric under its own weight [6]. Subsequent work extended such physical evaluation by examining static and dynamic drape behavior and its relationship to fabric characteristics [7].

In parallel with physical measurement, subjective and visual evaluations have been used to investigate how drape is perceived as part of garment appearance. Studies have examined the aesthetic impression of drape [8], the effects of material properties on gathered skirts and their visual evaluation [9], and the formation and visual characteristics of drape in flared-skirt models [10, 11]. Other studies have related the silhouettes of flared and gathered skirts to fabric mechanical, bending, and vibration properties [12, 13]. Together, these studies demonstrate that garment drape is not solely a physical material property but also a perceptual phenomenon influenced by garment structure and visual appearance.

The present study differs from this conventional line of textile research in that no physical fabric is measured. Instead, we examine the perceived drape of synthetically manipulated garment images generated by a diffusion model. Accordingly, drape intensity is used here as an operational perceptual term rather than as a substitute for conventional physical measures such as the drape coefficient. This distinction allows established knowledge of garment drape perception to be connected with the emerging problem of perceptual validation in generative fashion imagery.

B. Generative AI and Attribute Control in Fashion

Recent progress in image generation has shifted attention from the synthesis of realistic images toward controllable generation and editing. Diffusion models provide a flexible foundation for this transition by progressively reconstructing images from noisy representations [1]. Latent diffusion further reduces the computational burden of this process by performing generation in a compressed representation space, enabling practical high-resolution synthesis and editing [2]. Additional conditioning mechanisms have subsequently been introduced to provide more explicit control over generated content. For example, ControlNet incorporates structural conditions into diffusion-based generation [14], while instruction-guided approaches enable existing images to be modified according to natural-language descriptions [15].

Another line of research has investigated whether interpretable visual changes can be produced by manipulating internal representations of generative models. Studies based on GANs demonstrated that variations in attributes can be associated with directions or subspaces within learned latent representations. InterFaceGAN identified semantic boundaries that support controlled manipulation of facial attributes [16], whereas GANSpace derived interpretable controls from principal directions in intermediate latent representations [17]. StyleCLIP further demonstrated that language-based semantic information can be connected to latent editing operations [18]. Although these approaches differ in architecture and manipulation strategy, they collectively show that learned representations can support continuous and interpretable changes in generated images.

Such controllability is particularly relevant to fashion, where the usefulness of generative systems depends not only on image realism but also on the ability to preserve garment identity while modifying selected characteristics. Recent reviews describe applications of generative AI to fashion design, garment synthesis, virtual try-on, and image editing, while also identifying fine-grained controllability and evaluation as continuing challenges [3]. However, successful manipulation in a generative representation does not by itself establish that the intended attribute change is visually interpreted in the same way by human observers. For perceptual garment attributes such as drape, this distinction motivates an evaluation framework that considers not only the behavior of the generative representation but also the resulting human perceptual response.

C. Automated Evaluation and Human Perceptual Alignment

The increasing capability of generative models has created a corresponding need for scalable methods to evaluate generated images. Vision-language models provide one approach by representing visual and linguistic information within related feature spaces. CLIP, trained through contrastive learning on large-scale image–text pairs, has become a widely used representation for measuring semantic relationships between images and textual concepts [19]. Other approaches have moved beyond general image–text similarity toward models trained to approximate human preferences for generated images. Examples include ImageReward [20] and Human Preference Score v2 [4], which use human preference data to improve the correspondence between automated evaluation and subjective judgments.

The existence of such preference-oriented metrics also reflects an important limitation of purely automated evaluation: a systematic change in a learned representation is not necessarily equivalent to a systematic change in human perception. Deep perceptual metrics such as LPIPS have demonstrated that learned visual representations can capture aspects of perceptual similarity more effectively than simple pixel-level measures [21]. Nevertheless, perceptual judgments remain dependent on the task, the visual attribute under consideration, and the context in which that attribute appears. Consequently, automated metrics are useful for characterizing generated-image behavior but cannot automatically be treated as substitutes for task-specific human validation.

Human paired-comparison methods provide a complementary means of evaluating subtle perceptual differences without requiring participants to assign absolute numerical values to visual attributes. In a two-alternative forced-choice (2AFC) experiment, observers select one of two stimuli according to a specified perceptual criterion, allowing comparative judgments to be collected under a relatively simple task design [22]. Models for paired-comparison data, such as the Bradley–Terry model, can then transform these outcomes into latent worth estimates representing relative perceptual strength

[23]. This combination is particularly useful when the experimental question concerns the ordering of a continuously manipulated attribute rather than the assignment of an absolute physical quantity.

In our previous work, we combined latent-space diagnostics, CLIP-based evaluation, and human paired-comparison judgments to investigate perceptual reliability across multiple controlled attributes in diffusion-based fashion generation [5]. That study showed that agreement among the different evaluation layers varied across attributes, motivating closer examination of cases in which automated and perceptual assessments do not coincide. The present study takes a more focused approach by treating garment drape as a textile-oriented perceptual attribute and by examining correspondence at the level of individual base garments. This shift from multi-attribute aggregate validation to garment-specific analysis enables us to investigate whether apparently consistent automated trajectories can conceal stimulus-dependent differences in human perceptual ordering.

III. METHOD

A. Overview of the Semi-Automated Evaluation Pipeline

To evaluate whether latent-space attribute manipulation produces perceptually meaningful changes in garment drape, we developed a semi-automated evaluation pipeline that integrates AI-based stimulus generation, human perceptual evaluation, and automated feature-space evaluation into a unified experimental workflow. Rather than relying solely on either human judgment or automated metrics, the proposed pipeline enables systematic comparison between the two evaluation tracks, thereby providing a reproducible framework for assessing the perceptual validity of latent attribute manipulation.

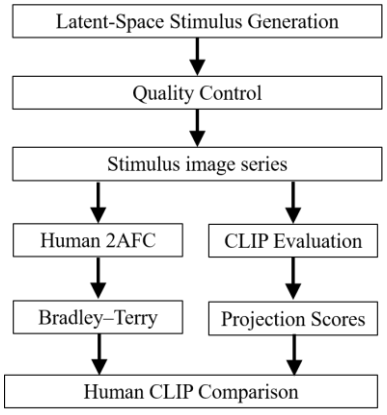


Fig. 1. Overview of the semi-automated evaluation pipeline. Human perceptual evaluation and CLIP-based feature-space evaluation are conducted as parallel tracks and subsequently compared to assess perceptual consistency.

As illustrated in Fig. 1, the pipeline consists of five functional stages: 1) stimulus generation by latent-space manipulation, 2) quality control of generated stimuli, 3) human perceptual evaluation using a web-based two-alternative forced-choice (2AFC) task, 4) CLIP-based automated feature-space evaluation, and 5) comparative analysis between Bradley–Terry perceptual scores and CLIP projection scores. Human evaluation and automated evaluation were performed independently and compared only during the final analysis stage.

B. Stimulus Generation

Stimulus images were generated using a Stable Diffusion-based inpainting pipeline with the Realistic Vision V5.1 Inpainting checkpoint (Realistic_Vision_V5.1-inpainting.safetensors). Generation was performed at a resolution of 512 × 512 pixels using 35 inference steps, a guidance scale of 7.5, and a denoising strength of 0.7. An initial mannequin image and a garment-region mask were used for inpainting. A randomly initialized seed was used for each generation and recorded in the corresponding generation metadata to enable traceability of the generated stimuli.

Five female garment images were manually selected from the generated candidates and used as base stimuli (Dress A–E). The selection was intended to provide visually distinct garment examples for this pilot evaluation; no formal probabilistic or stratified sampling procedure was applied. Accordingly, these five garments should be regarded as an exploratory stimulus set rather than as a representative sample of a broader garment population. The base condition was generated using the prompt "moderate drape, gentle folds clearly visible, slightly gathered at the waist." For each selected base garment, weak- and strong-drape endpoint images were subsequently generated using the same inpainting framework. The minimum-drape condition was specified using the prompt "studio product photo of a mannequin torso wearing a dress with no visible drape, smooth fitted surface, minimal folds, matte fabric, clean background," whereas the maximum-drape condition was specified as "studio product photo of a mannequin torso wearing a dress with strong drape, deep flowing folds, gathered at the waist, matte fabric, clean background." These descriptions served as operational anchors for the drape manipulation rather than as established physical definitions of minimum and maximum fabric drape.

Each generated endpoint image was encoded using the variational autoencoder (VAE) associated with the diffusion pipeline, and the resulting latent representation was stored after application of the model scaling factor. Thus, the present procedure did not employ a separate latent inversion algorithm such as DDIM inversion; instead, the generated images were re-encoded into the model's VAE latent space.

For each base garment, the stored minimum- and maximum-drape latent representations were used as the endpoints of the manipulation trajectory. Intermediate latent representations were generated by spherical linear interpolation (SLERP) between these two endpoints. Five equally spaced manipulation levels, $\alpha = -1.5$, -0.75, 0.0, 0.75, and 1.5, were used. The α values were mapped linearly onto the interpolation interval between the minimum- and maximum-drape latent representations, with $\alpha = -1.5$ and $\alpha = 1.5$ corresponding to the two endpoints and $\alpha = 0$ representing the midpoint of the interpolation trajectory. The interpolated latent representations were decoded using the VAE to produce the final 512 × 512 stimulus images. The resulting stimulus set comprised 25 images (5 base garments × 5 α levels). Representative images at $\alpha = -1.5$, 0.0, and 1.5 are presented in Fig. 2.

Before the final stimulus sequences were used for perceptual evaluation, the latent configuration was subjected to an automated quality-control procedure based on our previous evaluation framework [5]. For each base garment, the difference

between the distances from the base latent representation to the minimum and maximum endpoints (Δ) and the relative centrality of the base representation (c) along the minimum-to-maximum direction were calculated. The mean values of Δ and c across the five base garments were then used for quality-control classification. A configuration was classified as PASS when |mean Δ| < 8.0 and mean c > 0.3, and as WEAK-PASS when |mean Δ| < 12.0 and mean c > 0.15. Configurations not satisfying these conditions were classified as FAIL. These threshold values were empirically determined through iterative testing during the development of our previous evaluation framework [5] and were retained in the present study as practical quality-control criteria rather than theoretically derived universal thresholds.

Prior to deployment, all generated stimulus images were visually inspected by the authors. No images were excluded from the final stimulus set because all generated images passed this visual quality-control procedure without requiring regeneration.

C. Human Perceptual Evaluation

Fifteen participants completed the experiment. Because this study was conducted as a preliminary evaluation of the proposed pipeline, demographic information such as age, gender, and occupation was not collected. The experiment comprised ten primary comparison trials: two α intervals (−1.5 vs. 0.0 and 0.0 vs. 1.5) for each of the five base garments. One additional replicate trial of the Dress B comparison between α = 0.0 and α = 1.5 was included as a limited within-participant consistency check, resulting in 11 trials per participant. This replicate was intended only to provide a basic check of repeated responses for one comparison and was not designed as a comprehensive assessment of reliability across garments or comparison difficulty. Trial order was randomized, and the left-right presentation of each image pair was randomized. The experiment therefore yielded a total of 165 paired-comparison records.

D. CLIP-Based Feature-Space Evaluation

Following our previously proposed CLIP-based evaluation framework [5], automated feature-space evaluation was performed independently for each base garment using CLIP image embeddings. The evaluation was implemented using the OpenCLIP library with OpenAI-pretrained weights. CLIP was originally pretrained on large-scale image–text pairs; however, no text prompt was directly used to define the drape-related evaluation direction in the present analysis. Instead, as described below, the direction was defined separately for each base garment from the CLIP image embeddings of its minimum- and maximum-α endpoint images.

For each base garment, five generated images corresponding to the manipulation levels α = −1.5, −0.75, 0.0, 0.75, and 1.5 were encoded using the CLIP image encoder. Each resulting image embedding was L2-normalized before subsequent analysis.

The drape-related direction was defined separately for each base garment using the normalized CLIP image embeddings of the two endpoint stimuli. The minimum endpoint corresponds to α = −1.5, and the maximum endpoint corresponds to α = 1.5. Let $v_{b,min}$ and $v_{b,max}$ denote the normalized CLIP image embeddings of these two endpoint images, respectively. The normalized drape direction d_b was defined as follows:

$$d_b = \frac{(v_{b,max} - v_{b,min})}{\|v_{b,max} - v_{b,min}\|} \quad (1)$$

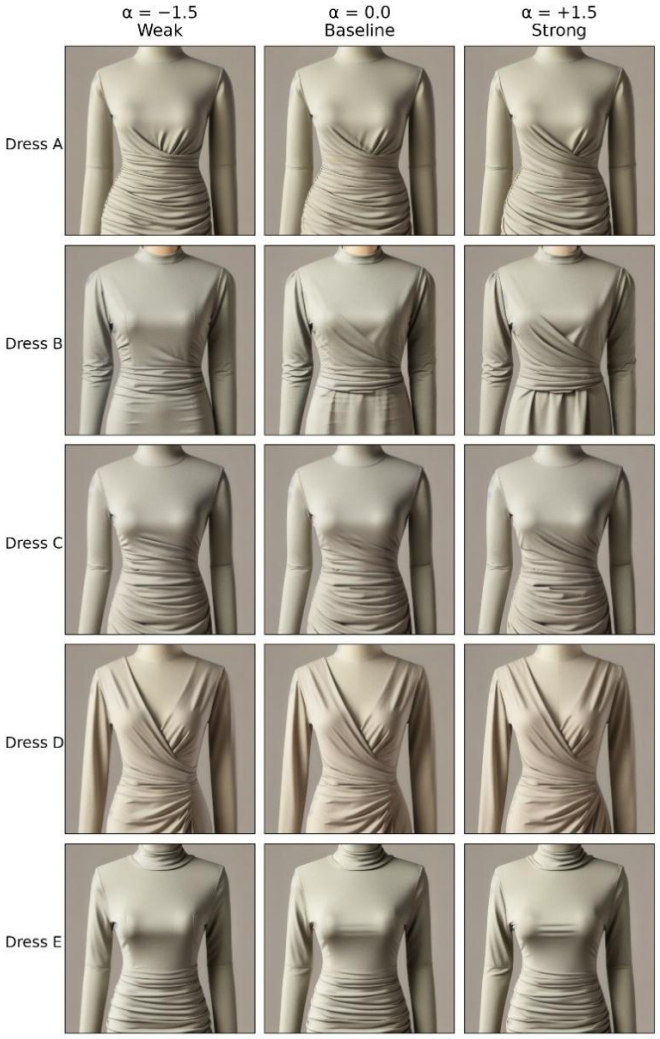


Fig. 2. Representative stimulus sequences for the five base garments (Dress A–E) at three drape manipulation levels: α = −1.5 (weak), 0.0 (baseline), and +1.5 (strong).

The midpoint between the minimum and maximum endpoint embeddings was then calculated. Let $v_{b,mid}$ denote this midpoint:

$$v_{b,mid} = \frac{v_{b,max} + v_{b,min}}{2} \quad (2)$$

For each generated image, its relative position along the drape-related direction was quantified using a CLIP projection score. Let $v_{b,\alpha}$ denote the normalized CLIP image embedding of the image generated from base garment **b** at manipulation level **α**. The CLIP projection score was calculated as follows:

$$S_{CLIP(b,\alpha)} = \left(v_{b,\alpha} - v_{b,mid}\right)^T d_b \quad (3)$$

The CLIP projection score represents the relative position of each generated image along the feature-space direction connecting the minimum- and maximum-α images of the same base garment. Importantly, this analysis does not assume that CLIP contains a predefined category corresponding to drape intensity. Instead, the drape-related direction is operationally defined for each base garment from the CLIP image embeddings of its two endpoint stimuli. Accordingly, the CLIP analysis should be interpreted as an internal feature-space consistency check of whether the intermediate stimuli progress systematically between the predefined endpoints, rather than as an independent semantic validation that CLIP recognizes drape intensity.

For each base garment, monotonic consistency between **α** and the CLIP projection score was evaluated using Spearman's rank correlation coefficient **ρ**. A positive value of **ρ** indicates that the CLIP projection score tends to increase as **α** increases.

In addition, trajectory straightness was calculated to quantify how directly the five image embeddings progressed from the minimum-α endpoint to the maximum-α endpoint in CLIP feature space. The straightness score was calculated as follows:

$$S_{str} = \frac{\|v_{b,max} - v_{b,min}\|}{\sum_{i=1}^{n-1} \|v_{b,i+1} - v_{b,i}\|} \quad (4)$$

A straightness score closer to **1** indicates that the sequence of image embeddings follows a more direct trajectory between the two endpoint embeddings.

E. Comparative Analysis

Human paired-comparison data were analyzed using the Bradley–Terry (BT) model to estimate perceptual worth scores β. The probability $P(i \succ j)$ that stimulus i (with coefficient α_i) is perceived as having stronger drape than stimulus j (with coefficient α_j) is defined as:

$$P(i \succ j) = \frac{\exp(\beta_i)}{\exp(\beta_i) + \exp(\beta_j)} \quad (5)$$

These perceptual worth scores were subsequently compared with CLIP projection scores S_{CLIP} to evaluate the agreement and potential divergence between human visual perception and automated feature-space evaluation across the manipulated drape levels.

IV. Results

A. Stimulus Quality and CLIP-Based Feature-Space Evaluation

The latent configuration satisfied the predefined PASS criteria described in Section III-B. The measured mean Δ and mean c values were −0.791 and 0.537, respectively, both comfortably satisfying the predefined PASS criteria (|mean Δ| < 8.0 and mean c > 0.3).

To examine whether the aggregate PASS classification concealed weaker configurations for individual garments, the same quality-control metrics were also examined separately for each base garment. Table I summarizes the garment-specific Δ and c values. All five garments had |Δ| < 8.0 and c > 0.3, with Δ ranging from −2.578 to 2.234 and c ranging from 0.388 to 0.596. Thus, all individual garments satisfied the corresponding numerical conditions used in the predefined aggregate QC criteria.

Visual inspection of the final stimulus set identified no images requiring regeneration or exclusion.

TABLE I. GARMENT-SPECIFIC LATENT-SPACE QUALITY-CONTROL METRICS

Dress	Δ	c
A	−1.803	0.596
B	−2.578	0.564
C	−0.649	0.544
D	−1.160	0.591
E	2.234	0.388
Mean	−0.791	0.537

Fig. 3 shows the CLIP projection scores along the base-specific directions defined by the endpoint image embeddings for the five base garments. For all five garments, the projection score increased monotonically as α increased from −1.5 to 1.5. Spearman's rank correlation coefficient between α and the projection score was ρ = 1.0 for all five image sequences.

Thus, all five generated image sequences exhibited monotonic trajectories along their respective base-specific CLIP projection directions over the evaluated α range.

B. Human Perceptual Evaluation

The 2AFC experiment yielded 165 paired-comparison records from 15 participants. The replicated Dress B comparison between α = 0.0 and α = 1.5 yielded consistent responses for 14 of the 15 participants (93.3%). Because this consistency check was based on a single replicated comparison, the value should not be interpreted as an estimate of overall within-participant reliability across the full experiment.

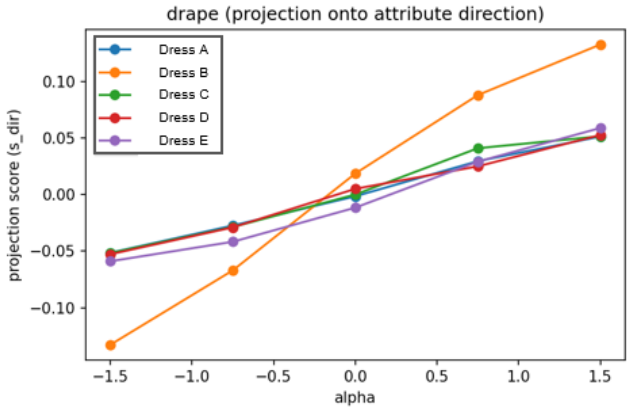


Fig. 3. CLIP-based projection scores along the drape attribute direction for five base images.

Fig. 4 shows the proportion of trials in which the higher-α stimulus was selected as having a stronger drape impression for each base garment and comparison interval. Across all base garments, the higher-α stimulus was selected in 42 of 75 comparisons (0.560) for α = −1.5 versus 0.0 and in 77 of 90 comparisons (0.856) for α = 0.0 versus 1.5.

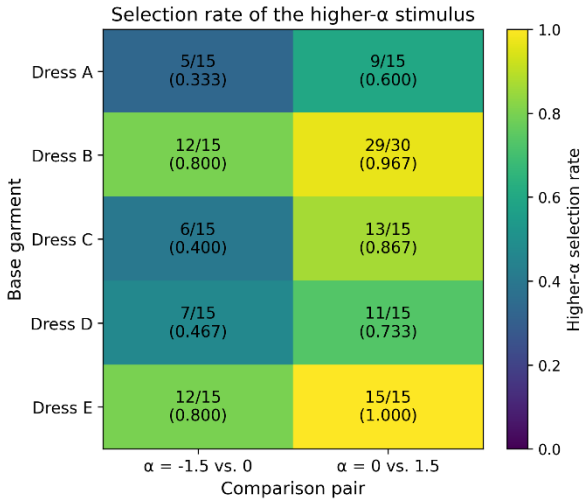


Fig. 4. Higher-α stimulus selection rates for each base garment and comparison interval in the 2AFC experiment. Each cell shows the number of higher-α selections over the number of comparisons, with the corresponding selection rate in parentheses.

The selection rates differed among the five base garments. For α = −1.5 versus 0.0, the higher-α selection rates were 0.333, 0.800, 0.400, 0.467, and 0.800 for Dresses A–E, respectively. For α = 0.0 versus 1.5, the corresponding rates were 0.600, 0.967, 0.867, 0.733, and 1.000. The neutral-to-high comparison for Dress B contained the additional replicate trial and therefore comprised 30 responses rather than 15.

The aggregated paired-comparison data were analyzed using the Bradley–Terry model described in Section III-E. The normalized BT worth values were 0.102, 0.130, and 0.768 for α = −1.5, 0.0, and 1.5, respectively, showing an aggregate monotonic ordering with α.

C. Garment-Specific Bradley–Terry Analysis

To examine variation in perceptual ordering among the individual base garments, the Bradley–Terry model was fitted separately to the paired-comparison data for each garment using the same estimation procedure as in the aggregate analysis. Table II summarizes the resulting normalized BT worth scores for the three α levels evaluated in the human perceptual experiment.

TABLE II. GARMENT-SPECIFIC BRADLEY–TERRY WORTH SCORES

Dress	α = −1.5	α = 0.0	α = 1.5
A	0.444	0.222	0.333
B	0.008	0.033	0.959
C	0.167	0.111	0.722
D	0.234	0.204	0.562
E	<0.001	<0.001	>0.999

For Dress E, the BT fit approached the boundary because the α = 1.5 stimulus was selected in all comparisons against α = 0.0; therefore, these values should be interpreted as indicating an extremely strong ordering toward α = 1.5 rather than as stable finite estimates.

The garment-specific BT worth scores differed substantially among the five base garments. Dress B exhibited a monotonic increase in BT worth across the three evaluated α levels, with values of 0.008, 0.033, and 0.959 for α = −1.5, 0.0, and 1.5, respectively. Dress E also showed a strong ordering toward the highest α level, with estimated worth values below 0.001 for α = −1.5 and 0.0 and above 0.999 for α = 1.5. As noted in Table II, the α = 1.5 stimulus for Dress E was selected in all comparisons against α = 0.0, resulting in a boundary-like BT estimate.

In contrast, the estimated BT worth scores for Dresses A, C, and D showed local reversals across the three evaluated α levels. For Dress A, the normalized worth values were 0.444, 0.222, and 0.333 for α = −1.5, 0.0, and 1.5, respectively. For Dress C, the corresponding values were 0.167, 0.111, and 0.722, and for Dress D they were 0.234, 0.204, and 0.562. Thus, α = 1.5 had the highest estimated worth for Dresses C and D, whereas the point estimate at α = −1.5 was slightly higher than that at α = 0.0. Dress A showed a different estimated ordering, with the highest worth at α = −1.5, followed by α = 1.5 and α = 0.0. Because these garment-specific BT worth values were estimated from a limited number of binary comparisons, the observed local reversals should be interpreted as descriptive, exploratory patterns rather than as statistically established violations of monotonicity.

V. DISCUSSION

A. Agreement and Divergence Between Automated and Human Evaluation

The results revealed both overall agreement and garment-specific divergence between the CLIP-based and human perceptual evaluations. In the automated evaluation, all five garment sequences exhibited monotonic CLIP projection scores across the five α levels, with Spearman's ρ = 1.0 for every base garment. Human evaluation also showed monotonic ordering at the aggregate level across the three α levels included in the human experiment, with normalized Bradley–Terry worth increasing from 0.102 at α = −1.5 to 0.130 at α = 0.0 and 0.768 at α = 1.5.

However, this aggregate agreement did not extend uniformly to the individual garments. The garment-specific BT analysis showed a clear monotonic pattern for Dress B and a strong ordering toward the highest α level for Dress E, whereas the estimated BT worth scores for Dresses A, C, and D contained local reversals across the three α levels included in the human experiment. Given the limited number of binary comparisons, these reversals should be regarded as exploratory garment-specific patterns rather than statistically established non-monotonic effects. In particular, the low-to-neutral comparison was less consistent than the neutral-to-high comparison: the higher-α stimulus was selected in 56.0% of the former comparisons and 85.6% of the latter comparisons.

These findings highlight a distinction between feature-space consistency and perceptual consistency. The CLIP-based analysis characterizes how generated images progress along a direction defined by the endpoint images in the model's feature space. Because the minimum- and maximum-α images themselves define this direction, their relative ordering is inherent in the evaluation procedure, and the resulting perfect rank correlations should not be interpreted as independent evidence that CLIP recognizes drape intensity. The informative result of this analysis is instead whether the intermediate images progress systematically and monotonically between the predefined endpoints. Such a monotonic feature-space trajectory,

however, does not necessarily imply that human observers perceive the corresponding images as an equally ordered progression of drape intensity. The garment-specific human analysis was therefore important because the aggregate results alone would have suggested substantially stronger agreement between automated and human evaluation.

B. Garment-Dependent Perceptual Effects

Although latent-space manipulation was used to vary drape with each base garment serving as a common reference, the differences among Dresses A–E suggest that the perceptual effect of the manipulation may depend on the visual context of the base garment. As illustrated by the stimulus examples in Fig. 2, the same range of α values produced visually different manifestations of drape across the five garments. Dresses B and E showed human perceptual ordering broadly consistent with increasing α, whereas the estimated BT worth scores for Dresses A, C, and D showed local reversals in at least part of the evaluated range. These patterns are descriptive and should be interpreted cautiously given the limited number of comparisons in this pilot study.

The present experiment does not establish which visual characteristics caused these garment-dependent differences or whether the manipulated perceptual impression can be attributed exclusively to drape. The operational endpoint descriptions intentionally referred to visually associated characteristics such as surface smoothness, fold depth, and gathering, and the resulting manipulation may therefore have covaried with silhouette, apparent fit, fold density, garment structure, or other visual properties. Participants were asked only to select which image exhibited a stronger sense of drape and did not independently rate these potentially correlated attributes. Accordingly, the present results should be interpreted as evaluating the operational perceptual construct of drape intensity defined in this study, rather than an isolated visual dimension that has been experimentally separated from all related garment attributes. Determining which visual cues contribute independently to perceived drape will require controlled attribute-specific evaluations in future work.

Importantly, perceptually ambiguous cases should be distinguished from generation failures. During quality control, stimuli were screened for clear generation failures and structural defects, but images were not excluded merely because the perceptual ordering of drape might be difficult to judge. Such cases are informative for the present research question because the purpose of the perceptual evaluation is precisely to determine whether technically controlled image changes produce consistent human judgments. Removing perceptually ambiguous stimuli in advance could therefore conceal cases in which automated control and human perception diverge.

C. Implications for Human-in-the-Loop Evaluation

The observed discrepancy illustrates the complementary roles of automated and human evaluation in generative fashion research. Automated feature-space analysis provides an efficient means of screening large numbers of generated image sequences and identifying systematic trajectories. In the present experiment, CLIP-based evaluation indicated monotonic progression for all five garment sequences. Human evaluation, in contrast, identified garment-specific deviations that were not apparent from the automated results alone.

This complementary relationship is central to the proposed semi-automated evaluation pipeline. The role of automation is not to replace human perceptual judgment but to reduce the effort required to generate, organize, screen, and quantitatively characterize controlled stimuli. Human evaluation can then be concentrated on determining whether technically consistent manipulations correspond to the intended perceptual effect. The pipeline therefore provides a practical mechanism for detecting cases in which model-space consistency does not translate directly into perceptual consistency.

The results also suggest a possible feedback loop for improving automated evaluation models. Garment–attribute combinations for which automated trajectories and human perceptual ordering disagree can be identified systematically and accumulated as perceptually informative cases. Such human-annotated discrepancies could potentially be used in future work to evaluate or adapt vision-language models for fashion- and textile-specific perceptual attributes. Rather than assuming that a general-purpose representation such as CLIP fully captures these attributes, the Human-in-the-Loop framework provides empirical data for determining where additional domain adaptation may be beneficial.

D. Limitations and Future Directions

Several limitations should be considered when interpreting these findings. First, this pilot study included only five AI-generated female garments and 15 participants. Because the base garments were manually selected rather than sampled according to predefined objective criteria, the present findings should not be interpreted as estimating the prevalence of such perceptual discrepancies across garment populations. The garment-specific patterns and local reversals in the estimated BT worth scores observed in the present analysis should therefore be interpreted as exploratory findings rather than as stable estimates of garment-dependent perceptual ordering. The observed garment-specific patterns cannot yet be generalized to other garment categories, designs, materials, or populations. Participant demographic information was not collected, which also prevents examination of whether perceptual judgments differed according to age, gender, fashion expertise, or other participant characteristics. In addition, the operational drape manipulation was not experimentally separated from potentially correlated visual attributes such as silhouette, apparent fit, fold density, gathering, or garment structure. Independent attribute ratings or controlled manipulations will be needed to determine the extent to which these visual factors contribute separately to perceived drape intensity.

In addition, each garment–α condition was represented by a single generated stimulus. Although the generation parameters and random seeds were recorded to ensure traceability, the present design does not distinguish systematic effects of the latent manipulation from variation associated with individual stochastic generations. The observed feature-space trajectories and perceptual judgments should therefore be interpreted as applying to the specific stimulus realizations examined in this pilot study. Future work should generate multiple independent realizations for each garment–α condition and evaluate the

robustness of both automated feature-space trajectories and human perceptual ordering across generation seeds.

The present Bradley–Terry analyses also treat the paired-comparison responses as aggregated outcomes and do not explicitly model participant-level heterogeneity or within-participant dependence across repeated judgments. Given the small number of participants and the sparse comparison structure of this pilot experiment, the garment-specific BT estimates were used primarily as descriptive summaries of perceptual ordering rather than as population-level inferential estimates. Future studies with larger samples and denser repeated-measures designs should employ hierarchical Bradley–Terry models or mixed-effects logistic regression to jointly account for participant- and garment-level variation.

Second, although five α levels were generated and evaluated using CLIP, the human 2AFC experiment evaluated only two intervals: $\alpha = -1.5$ versus 0.0 and $\alpha = 0.0$ versus 1.5. The present results therefore do not establish the detailed functional relationship between α and perceived drape intensity. In particular, the difference in selection rates between the two evaluated intervals should not be interpreted as evidence of a general nonlinear psychophysical response. The sparse comparison design also limits the stability of garment-specific Bradley–Terry estimates, particularly in cases of highly consistent choices that produce boundary-like fits. Future experiments should include a denser set of pairwise comparisons across the manipulation range.

Third, the automated evaluation used base-specific CLIP directions defined from the minimum- and maximum-α endpoint images. This procedure is useful as an internal feature-space consistency check for characterizing the relative trajectory of each generated sequence, but it does not provide an independent semantic validation of drape intensity or establish that CLIP contains an intrinsic or general representation of this attribute. Future work should therefore evaluate intermediate stimuli using independently defined directions, such as text-derived, externally learned, or held-out directions, and investigate whether fashion- or textile-specific representations provide closer correspondence with human perceptual judgments.

Finally, the present study examined only drape intensity as an operational perceptual attribute. Extending the same experimental framework to additional garment attributes, such as material appearance, glossiness, fit, or design-related variations, would help determine whether the observed automated–human discrepancies are specific to drape or represent a more general characteristic of fine-grained attribute control in generative fashion systems. Larger and more diverse perceptual datasets would also enable systematic investigation of domain-specific adaptation or additional training of vision-language models using human perceptual annotations.

In addition, the within-participant consistency check was based on only one replicated comparison (Dress B, $\alpha = 0.0$ vs. 1.5), which showed a relatively decisive selection pattern. The observed 93.3% agreement therefore cannot be generalized to more ambiguous comparisons or interpreted as an overall measure of response reliability.

VI. Conclusion

This study investigated the perceptual validity of latent-space control of garment drape intensity in AI-generated garment images using a semi-automated Human-in-the-Loop evaluation pipeline. Five base garments were manipulated across five α levels and evaluated independently using CLIP-based feature-space analysis and a human 2AFC experiment with Bradley–Terry scaling.

The CLIP-based evaluation showed monotonic trajectories for all five garment sequences, whereas human evaluation revealed a more complex pattern. Although the aggregate Bradley–Terry scores increased monotonically with α, garment-specific analysis revealed local reversals in the estimated perceptual ordering for three of the five base garments. Given the limited number of binary comparisons, these garment-specific reversals should be interpreted as exploratory patterns rather than as statistically established non-monotonic effects. These findings demonstrate that aggregate agreement between automated and human evaluation can mask stimulus-specific perceptual discrepancies and that feature-space consistency does not necessarily imply perceptual consistency.

The proposed framework provides a practical means of combining automated screening with human perceptual validation for fine-grained attributes in generative fashion systems. Future work will extend the evaluation to larger and more diverse garment and participant sets, additional perceptual attributes, and denser manipulation levels. The Human-in-the-Loop data obtained through such evaluations may also support the development or adaptation of automated models that better reflect fashion- and textile-specific human perception.

Declaration on Generative AI

Generative AI tools, including ChatGPT (OpenAI) and Gemini (Google), were used during manuscript preparation to assist with language refinement, organization, and improvement of clarity. All AI-assisted content was critically reviewed, revised, and verified by the authors. The scientific ideas, experimental design, data analysis, interpretation of the results, and conclusions presented in this study were developed and validated by the human authors, who take full responsibility for the content of the manuscript.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10684–10695, 2022.

[3] W. Shi, W. Wong, and X. Zou, "Generative AI in Fashion: Overview," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 4, Art. no. 74, 2025, doi: 10.1145/3718098.

[4] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li, "Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis," *arXiv preprint arXiv:2306.09341*, 2023, doi: 10.48550/arXiv.2306.09341.

[5] N. Kuwahara, S. Kawanami, T. Sato, and D. Choi, "Evaluating perceptual reliability of latent attribute control in diffusion-based fashion generation," *International Journal of Advanced Computer Science and*

Applications, vol. 17, no. 4, pp. 231–240, 2026, doi: 10.14569/IJACSA.2026.0170424.

[6] C. C. Chu, C. L. Cummings, and N. A. Teixeira, "Mechanics of elastic performance of textile materials: Part V: A study of the factors affecting the drape of fabrics—The development of a drape meter," *Textile Research Journal*, vol. 20, no. 8, pp. 539–548, 1950, doi: 10.1177/004051755002000802.

[7] M. Matsudaira and M. Yang, "Features of conventional static and new dynamic drape coefficients of woven silk fabrics," *Textile Research Journal*, vol. 73, no. 3, pp. 250–255, 2003, doi: 10.1177/004051750307300309.

[8] I. Uchiyama, T. Asai, and H. Shigeno, "Quantitative consideration on the beauty of drape," *J. Japan Research Association for Textile End-Uses*, vol. 15, no. 5, pp. 170–174, 1974, doi: 10.11419/senshoshi1960.15.170. (in Japanese)

[9] K. Tsuji, K. Ito, and N. Kato, "Effect of fabric properties on gathered skirts and their visual evaluation," *Journal of Home Economics of Japan*, vol. 32, no. 6, pp. 463–471, 1981, doi: 10.11428/jhej1951.32.463. (in Japanese)

[10] N. Suda and M. Ohira, "On the drapeability of fabrics: Visual evaluation of drapeability," *J. Japan Research Association for Textile End-Uses*, vol. 14, no. 4, pp. 122–126, 1973, doi: 10.11419/senshoshi1960.14.122. (in Japanese)

[11] N. Suda and M. Ohira, "On the drapeability of fabrics: Analysis of drape formation mechanism using a flared-skirt-shaped model," *J. Japan Research Association for Textile End-Uses*, vol. 20, no. 4, pp. 148–152, 1979, doi: 10.11419/senshoshi1960.20.148. (in Japanese)

[12] M. Miki, M. Ayada, and M. Niwa, "Effects of the mechanical properties of fabrics and the number of gore panels have on the silhouettes of flared skirts," *Journal of Home Economics of Japan*, vol. 46, no. 7, pp. 671–682, 1995, doi: 10.11428/jhej1987.46.671. (in Japanese)

[13] K. Izumi and M. Niwa, "Relation between the silhouette of gathered skirts in wear and bending, vibration properties of fabrics," *Journal of Home Economics of Japan*, vol. 34, no. 2, pp. 96–104, 1983, doi: 10.11428/jhej1951.34.96. (in Japanese)

[14] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 3836–3845, 2023.

[15] T. Brooks, A. Holynski, and A. A. Efros, "InstructPix2Pix: Learning to follow image editing instructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[16] Y. Shen, G. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9240–9249, 2020.

[17] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering interpretable GAN controls," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 9841–9850, 2020.

[18] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-driven manipulation of StyleGAN imagery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 2085–2094, 2021, doi: 10.1109/ICCV48922.2021.00210.

[19] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 8748–8763, 2021.

[20] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, F. Tang, and Y. Dong, "ImageReward: Learning and evaluating human preferences for text-to-image generation," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023.

[21] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 586–595, 2018.

[22] M. Jogan and A. A. Stocker, "A new two-alternative forced choice method," *Journal of Vision*, vol. 14, no. 3, 2014.

[23] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, nos. 3–4, pp. 324–345, 1952.

SGRL: A Hybrid Semantic-Graph Representation Learning Framework for Software Vulnerability Detection

Cong Bui Van
Posts and Telecommunications Institute of Technology (PTIT),
122 Hoang Quoc Viet, Nghia Do Ward, Hanoi City, Vietnam

***Abstract*—The increasing scale and complexity of software increase the risk of security vulnerabilities, creating a need for automated and effective detection methods. Transformer-based methods can learn semantic representations but do not fully exploit structural information, whereas graph neural network-based methods remain limited in representing token-level semantics. In addition, data imbalance affects models' learning and generalization capabilities. In this study, the author proposes SGRL (Semantic–Graph Representation Learning), a framework that combines CodeT5 and a Graph Convolutional Network (GCN) over a Code Property Graph (CPG) to jointly exploit the semantic and structural features of source code. The two feature sources are fused through concatenation and adjusted using BDC (Bernoulli Dropout in CNN) before classification by a Multi-Layer Perceptron (MLP). Experiments on the Verum dataset with the optimal configuration of a 512-token input length and a BDC probability of 0.2 achieve an Accuracy of 84.23%, Precision of 84.55%, Recall of 84.23%, and F1-score of 84.14%. Results on Verum and FFmpeg+Qume also show that SGRL outperforms REVEAL, Russell, VulDeePecker, SySeVR, and Devign under the corresponding experimental settings.**

***Keywords*—Source code vulnerability detection; CodeT5; graph convolutional network; code property graph; representation learning; deep learning**

I. Introduction

A. Problem Statement

In the context of digital transformation and the development of large-scale software systems, ensuring source code security is essential to minimizing the risk of vulnerability exploitation and system compromise. Programming errors such as buffer overflow, use-after-free, null pointer dereference, and out-of-bounds write can introduce serious vulnerabilities and may even lead to remote code execution [1]. Accordingly, source code vulnerability detection generally involves two main tasks: 1) extracting vulnerability-related indicators and features; and 2) analyzing these features to determine the presence of vulnerabilities [1–5]. On this basis, current source code vulnerability detection methods can be divided into two main categories: 1) rule- and pattern-based methods; and 2) machine learning- and deep learning-based methods.

1) Rule-based detection: Rule-based methods identify vulnerabilities using rules, patterns, or indicators of unsafe behavior derived from expert knowledge and known vulnerabilities. Common techniques include static analysis, pattern matching, reverse engineering, and testing. These methods have the advantages of interpretability and the ability to detect predefined types of vulnerabilities effectively. However, constructing and updating rules requires considerable expert effort; their generalization capability is limited, they are prone to false alarms, and they have difficulty detecting new types of vulnerabilities.

2) Machine learning- and deep learning-based detection: Machine learning and deep learning methods transform source code into feature representations to learn the differences between vulnerable and secure source code. The development of Transformer models such as BERT [6], RoBERTa [7], CodeBERT [8], and CodeT5 [9] has improved the ability to learn semantic and contextual representations of source code through relationships among tokens. However, these models primarily represent source code as sequences and do not directly model structural relationships such as syntax, control flow, and data dependencies.

Conversely, Graph Neural Network (GNN)-based methods can exploit program structure through graphs such as the Abstract Syntax Tree (AST) [10], Control Flow Graph (CFG) [11], Program Dependence Graph (PDG) [12], and Code Property Graph (CPG) [13]. In particular, CPG combines syntactic, control-flow, and data-flow information, enabling the model to learn dependencies among program components more comprehensively. GCN is a common GNN architecture capable of aggregating information from neighboring vertices to learn structural representations of graphs.

Nevertheless, most existing studies focus separately on semantic representation using Transformers or structural representation using GNNs, and therefore do not fully exploit the interaction between these two types of information. Furthermore, vulnerability detection data often exhibit considerable class imbalance between secure and vulnerable samples [14], increasing bias toward the majority class and affecting the ability to detect vulnerable samples. Consequently, combining semantic representations from Transformers, structural representations from GNNs, and a mechanism for mitigating the effects of data imbalance is a necessary approach to improving the effectiveness and generalization capability of source code vulnerability detection systems.

B. Research Motivation and Objectives

1) Research motivation: In recent years, deep learning methods for source code vulnerability detection have primarily developed along two directions: 1) semantic representation learning using Pre-trained Language Models (PLMs), and 2) program structural representation learning using Graph Neural Networks (GNNs) [15–18]. Models such as CodeBERT, GraphCodeBERT, CodeT5, and UniXCoder have significantly improved the semantic representation of source code. In particular, CodeT5 can learn contextual representations of tokens and statements, but it primarily processes source code as sequences and therefore does not fully exploit structural relationships such as control flow and data dependencies.

Conversely, methods based on AST, CFG, PDG, and particularly CPG combined with GCN can effectively exploit the program's structural relationships. However, the effectiveness of GCN depends substantially on the quality of the input vertex features. Methods using One-Hot Encoding, Word2Vec, FastText, or GloVe do not fully capture the contextual semantics of source code. Moreover, data imbalance remains a considerable challenge because vulnerability detection datasets often contain substantially more secure than vulnerable samples. Techniques such as Random Oversampling, SMOTE, ADASYN, and GAN can reduce imbalance but still entail risks of generating redundant samples, distorting data distributions, or causing overfitting.

These limitations indicate a remaining research gap in jointly exploiting deep semantic representations, program structural representations, and mitigation of data imbalance during representation learning. Therefore, this study proposes an architecture combining a Transformer and GCN, in which CodeT5 learns semantic features, CPG–GCN learns structural features, and a feature-rebalancing mechanism based on BDC is applied to improve generalization and source code vulnerability detection performance.

2) Proposed objectives and solution direction: To address the limitations of existing methods, this study proposes the SGRL Framework, a representation learning architecture that combines a Transformer and a graph neural network to exploit both semantic and structural information in source code. CodeT5 learns semantic representations from normalized source code, whereas CPG combined with GCN learns program structural representations. The two feature sources are then fused to construct a unified representation space for vulnerability detection.

Specifically, source code is first normalized to reduce noise and remove unnecessary components such as comments, redundant whitespace, and symbols that have no value for the representation process. The normalized source code is fed into CodeT5 to obtain fixed-dimensional semantic representations. Simultaneously, the source code is converted into a Code Property Graph (CPG), in which vertices represent program components and edges describe syntactic, control-flow, and data-flow relationships. Graph features are propagated and aggregated through GCN layers to learn program structural representations.

C. Principle of Source Code Vulnerability Detection of the SGRL Framework

Recent studies indicate that exploiting semantic or structural features in isolation is insufficient for a comprehensive source code representation because programs simultaneously contain semantic, syntactic, control-flow, and data-flow information. Therefore, combining the two types of representations is necessary to improve vulnerability detection effectiveness.

On this basis, this study proposes the SGRL Framework, which combines CodeT5 and GCN to jointly learn semantic and structural features of source code. CodeT5 extracts semantic representations, whereas GCN learns structural representations from CPG. The two feature vectors are fused by concatenation to form a combined representation space used as input to the vulnerability classification stage.

D. Contributions of the Study

Based on the limitations of existing methods, this study proposes the SGRL Framework, which combines a pre-trained language model and a graph neural network to jointly exploit semantic and structural features while mitigating the effects of data imbalance. The main contributions are as follows:

First, an architecture combining CodeT5 and GCN is proposed to jointly learn semantic features from source code and structural features from CPG, thereby constructing a more comprehensive representation for vulnerability detection.

Second, a parallel representation learning process is developed in which CodeT5 learns context-rich semantic representations, whereas GCN learns structural relationships through CPG, including syntactic, control-flow, and data-flow information.

Third, a feature-fusion mechanism based on concatenation is proposed to preserve both representation sources before they are passed to the classifier, thereby enhancing the ability to exploit relationships between semantic and structural information.

Fourth, this study proposes a new data-balancing method called BDC, based on generating data for the minority class through the Bernoulli distribution in the Dropout layer of a CNN. BDC balances the minority and majority classes in the experimental dataset. Applying BDC after feature fusion aims to enhance the model's generalization capability under imbalanced-data conditions. Unlike traditional data generation methods, BDC acts as an overfitting-reduction technique: this study exploits random feature removal according to a Bernoulli distribution to generate multiple representation combinations during training. This approach helps limit the model's dependence on a few dominant features while improving representation learning for minority-class samples without using synthetic sample generation techniques such as SMOTE [19] or GAN [20].

Fifth, experiments are conducted on the Verum dataset using different token-length and BDC-probability configurations. The best result is obtained with 512 tokens and BDC = 0.2, yielding an accuracy of 84.23%, Precision of 84.55%, Recall of 84.23%, and an F1-score of 84.14%, demonstrating the effectiveness of combining semantic–structural representations with a feature-adjustment mechanism for source code vulnerability detection.

II. Related Works

A. Rule-Based Source Code Vulnerability Detection

According to Chen et al. [21], rule-based vulnerability detection is a traditional approach commonly used in static source code analysis tools. This method constructs rules or source code patterns from expert knowledge and known vulnerability types, and then compares source code against the rule set to identify indicators of potential information security risks [22]. Representative tools such as Flawfinder, Cppcheck, Clang Static Analyzer, SonarQube, and Fortify SCA typically rely on secure coding rules and knowledge bases such as CWE [23] and CERT C/C++, enabling detection of several common vulnerability types, including buffer overflows, invalid pointer use, and memory leaks.

In addition, Liu et al. [3] proposed VulMatch to detect recurring vulnerabilities (1-day and N-day) by combining source code and binary information. Experiments on open-source projects showed that VulMatch performed better than some binary matching methods, such as Asm2vec [24] and Palmtree [25], while also providing explainable detection results. However, the method still encounters difficulties when processing large-scale, complex source code or incompletely labeled data.

In general, rule-based methods offer interpretability and effectiveness for known vulnerability types, but are limited in generalization and the detection of new vulnerability types. Detection effectiveness depends substantially on the completeness of the rule set and typically requires manual updates when new vulnerability types or programming techniques emerge. Furthermore, rules are primarily based on syntactic patterns and therefore do not fully exploit program semantics, control flow, and data flow, resulting in risks of false alarms and missed vulnerabilities, particularly for source code with complex structures. These limitations have encouraged the development of machine learning and deep learning methods capable of automatically learning representations and generalizing vulnerability features from data.

B. Source Code Vulnerability Detection Based on Natural Language Processing

Advances in machine learning and deep learning have introduced new approaches to source code vulnerability detection, enabling models to learn features automatically from data rather than relying on manually constructed rules. Early methods employed Bag-of-Words (BoW), TF-IDF, Word2Vec [26, 27], and FastText in combination with algorithms such as SVM, Random Forest, or XGBoost. However, these representations primarily reflect statistical relationships among tokens and do not fully exploit contextual information or dependencies within programs.

The emergence of source code language models such as CodeBERT, CodeGPT, and CodeT5 has substantially improved semantic representation learning. Ding et al. [28] showed that evaluation results for source code language models may be affected by duplication and data leakage, while these studies have not fully addressed complex structures and relationships beyond the scope of a single function. Kalouptsoglou et al. [29] further investigated transfer learning with CodeGPT and CodeBERT, showing that token-level contextual representations are effective; however, these methods still primarily exploit sequential representations and do not directly integrate structural information such as control flow and data dependencies.

Recently, Zhang et al. [30] proposed CodeSAGE, which combines Code Attribute Graph–GraphSAGE with BiLSTM and Attention to jointly exploit structural and semantic features. Although positive results were achieved, the method has not exploited source code-specific pre-trained language models such as CodeT5 or directly fused semantic representations from Transformers with structural representations from CPG/GCN. These limitations highlight the need to develop models capable of deeper integration between program semantics and structure, providing a basis for the approach proposed in this study.

C. Source Code Vulnerability Detection Based on Graph Structure

To overcome the limitations of methods that exploit only sequential representations, numerous studies have shifted toward graph-based source code representations, particularly AST, CFG, PDG, and CPG. AST describes syntactic structure; CFG represents control flow; and PDG models data and control dependencies. Yamaguchi et al. [13] proposed the Code Property Graph (CPG), which jointly models syntactic information, control flow, and data dependencies within a unified graph structure, thereby providing a more comprehensive representation of program structure and behavior.

On this basis, GNN architectures such as GCN, GGNN, GAT, and Graph Transformer have been applied to learn structural representations for vulnerability detection. Liu et al. [31] proposed Vul-LMGNNs, reporting positive results on the REVEAL and SARD datasets compared with Devign [32], LineVD [33], and FUNDED [34]. Subsequent studies, including iGnnVD [35], Adaptive GNN [36], and Heterogeneous GNN [37], further exploited diverse relationships and features in program graphs. In addition, VulMPFF [38] combines source code sequence information, lexical–syntactic relationships, and graph structure, achieving improved results over several existing methods. Studies by Cong et al. [39, 40] also indicate that combining Transformers and GNNs has the potential to improve representation learning by jointly exploiting semantic and structural source code information.

However, GNN-based methods still exhibit several limitations. First, representation learning effectiveness depends considerably on the quality of initial node features; methods using One-Hot Encoding, Word2Vec, or GloVe do not fully reflect the contextual semantics of source code. Second, many studies focus on graph-based information propagation without effectively exploiting pre-trained language models to enhance node-level semantic representations. Consequently, such models may capture structural relationships well but remain limited in distinguishing code segments with similar structures but different semantics and security properties. This indicates the need to combine pre-trained language models with GNNs to construct more comprehensive semantic–structural representations for source code vulnerability detection.

III. PROPOSED MODEL

A. Architecture of the SGRL Model

Overall, the SGRL Framework consists of five main stages: 1) source code normalization; 2) semantic representation learning; 3) structural representation learning; 4) feature fusion and rebalancing; and 5) vulnerability classification. The entire processing pipeline is illustrated in Fig. 1.

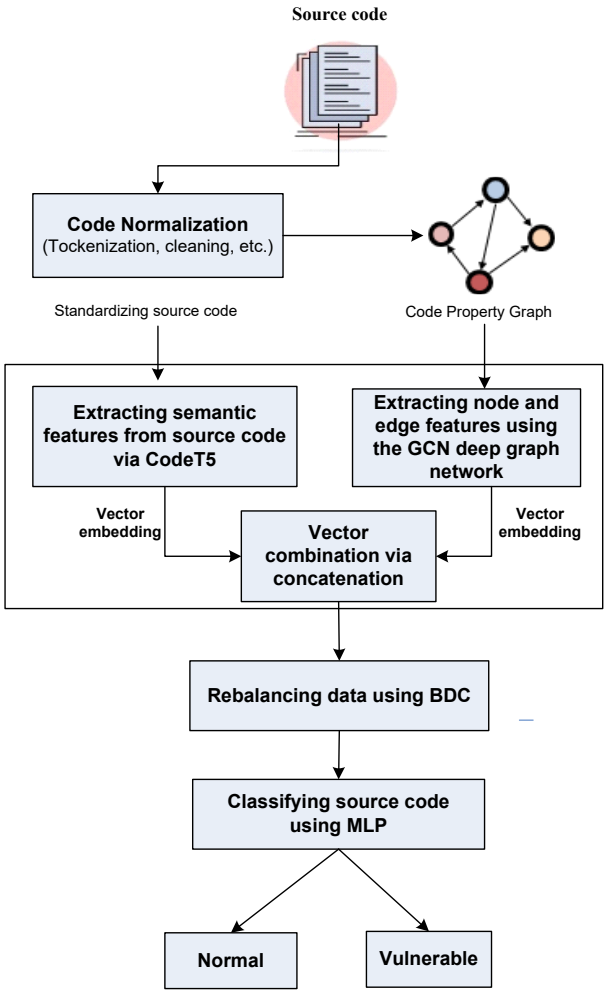


Fig. 1. Overall architecture of the SGRL framework.

The functions, tasks, and operating process of the blocks in the SGRL model are as follows:

- "Source code": The source code dataset used is Verum [41], written in C/C++.
- "Code Normalization" block: Removes redundant data and comments, searches for and checks erroneous source code segments, and removes them from the source code.
- "Code Property Graph" block: Analyzes and transforms the source code into CPG form using the Joern tool [42]. Joern analyzes source code in terms of syntax and semantics to represent it as a CPG graph.
- "Extracting semantic features from source code via CodeT5" block: Extracts semantic features of the source code, including features of individual nodes or of the entire code segment.
- "Extracting node and edge features using the GCN deep graph network" block: Extracts source code features from the CPG graph.
- "Vector combination via concatenation" block: Fuses two feature vectors—semantic source code features extracted using CodeT5 and structural features extracted from the graph using GCN—through concatenation.
- "Rebalancing data using BDC" block: After the source code feature profiles are obtained, the authors observe a very large difference between the numbers of normal source code features and vulnerable source code features. Immediate classification may result in inaccurate predictions or, in some cases, model overfitting. Therefore, this study proposes generating data from the minority class to balance the experimental dataset using BDC. The result is a new dataset containing source code feature profiles balanced between the vulnerable and non-vulnerable labeled classes.
- "Classifying source code using MLP" block: Detects and classifies source code vulnerabilities based on the features constructed by the "Vector combination via concatenation" block. The author proposes using MLP for source code vulnerability classification.

The above describes the operation of the blocks in Fig. 1. The following sections describe the model's source code vulnerability detection task in detail.

B. Data Preprocessing

1) Source code normalization: In the first stage, source code is preprocessed and normalized to remove redundant components such as comments, whitespace, and unnecessary characters; at the same time, variable names, function names, and parameters are standardized according to a unified rule. This process helps reduce vocabulary diversity, limit sparse data, and improve the model's generalization capability. Preprocessing is necessary because raw source code contains many symbols, complex syntax, and differences in programming style, making the representation learning process difficult.

2) Representing source code as a CPG graph:

a) Introduction to CPG: CPG, introduced by Fabian Yamaguchi et al. in 2014 [13], combines AST, CFG, and PDG into a common data structure consisting of nodes and node types representing source code components such as classes and methods; labeled directed edges representing relationships among components; and key–value pairs containing node attributes, where valid keys depend on the node type. In this study, the author proposes normalizing and converting source code into CPG form to facilitate feature extraction.

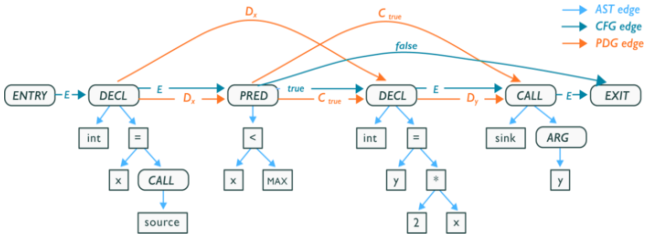


Fig. 2. Representation of source code according to CPG.

Fig. 2 shows that CPG is the combination of AST, CFG, and PDG, helping provide more important information for source code feature extraction because these graphs describe both static features (variable names, functions, etc.) and their relationships.

b) Introduction to the Joern platform: Joern [43] is a platform commonly used today in static analysis for source code, including C/C++ source code. It enables vulnerability detection by converting source code into a CPG through nodes (functions, statement blocks, etc.) to describe source code features. This study applies the Joern tool to represent C/C++ source code as a CPG graph consisting of nodes containing syntactic code snippets and syntactic edges (AST), control edges (CFG, PDG), thereby expressing all properties of the source code.

C. Source Code Feature Extraction

From Fig. 1, combining representation learning using a Transformer model and a graph neural network to jointly exploit semantic and structural source code information is an approach intended to maximize the use of different source code feature components. After normalization, the source code is processed in parallel along two directions: 1) the first branch passes it through CodeT5 to learn semantic representations at the token and statement levels; 2) the second branch represents the normalized source code as a CPG graph and then learns graph features using GCN. Regarding the second branch, studies [13, 40, 41, 43] have listed the vertex and edge features of CPG. CPG is a particularly useful representation of original code because it provides a compact, integrated representation comprising elements of control-flow and data-flow graphs, in addition to AST and the Program Dependence Graph (PDG). Each of these elements provides additional context about the overall semantic structure of the code [13]. Formally, CPG is denoted by G = (V, E), where V represents graph vertices (or nodes) and E represents edges. Each vertex in CPG comprises a vertex type (e.g., ArithmeticExpression or CallStatement) and a segment of original code. Each edge contains information about its edge type (e.g., IS_AST_PARENT or FLOWS_TO) [42].

The semantic and structural features extracted by CodeT5 and GCN, respectively, are fused through concatenation to form a combined feature vector for the subsequent classification stage. This is an entirely new proposal for source code feature extraction that uses a model combining semantic and structural representation learning, rather than focusing on a single feature type as is currently common. With this new proposal, important source code features can be extracted more comprehensively. The following sections describe the source code feature extraction process based on CodeT5 and GCN in detail.

1) Extracting semantic features of source code using CodeT5

a) Overview of CodeT5: In 2021, Wang et al. proposed CodeT5 [9], an Encoder-Decoder Transformer model pre-trained specifically for source code. Unlike previous models such as CodeBERT, which use only an Encoder architecture, CodeT5 is designed to support both code understanding and code generation tasks through a unified architecture. In particular, CodeT5 introduces an Identifier-aware Pre-training mechanism, allowing the model to learn the semantic roles of identifiers in programs, thereby significantly improving the quality of source code representation across many tasks such as bug detection, vulnerability detection, clone detection, and code generation.

After that, CodeT5 uses the Multi-Head Self-Attention mechanism to model relationships among all tokens in the code segment. For each attention head, the three matrices Query, Key, and Value are computed as follows:

$$Q = X\mathrm{W}_Q, K = X\mathrm{W}_K, V = X\mathrm{W}_V, \quad (1)$$

where, WQ, WK, and WV are weight matrices to be learned. The attention weight is determined according to Eq. (2) below:

$$Attention(Q,K,V) = Soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2)$$

where, dk is the dimension of the key vector.

b) Method for extracting semantic features of source code using CodeT5: After normalization, the source code is simultaneously fed into two independent representation learning branches. The first branch uses CodeT5 to exploit the semantic features of the program. In this branch, CodeT5, a Transformer model pre-trained on a large volume of multilingual source code, can learn contextual representations that reflect the meaning of each token as well as the relationships among statements in the program. The output of CodeT5 is a fixed-dimensional semantic representation vector that reflects the content and context of the entire source code segment.

It can be said that the process of extracting semantic features of source code using CodeT5 is described in the following figure:

From Fig. 3, it can be seen that the semantic feature extraction process is performed through the following steps:

Step 1: After normalization, each code segment is represented as a token sequence according to Eq. (3) below:

$$S = \{t_1, t_2, ..., t_L\}, \quad (3)$$

Step 2: Here, L is the number of tokens after normalization. The token sequence is fed into the encoder of CodeT5. Each token is first mapped into an embedding vector in a d-dimensional feature space, as shown in Eq. (4):

$$E = \{e_1, e_2, \ldots, e_L\}, e_i \in \mathbb{R}^d. \quad (4)$$

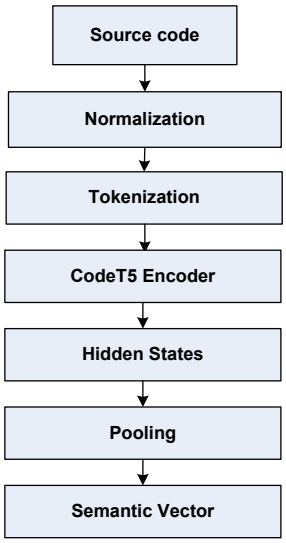


Fig. 3. The process of extracting semantic features of source code using CodeT5.

Step 3: To preserve information about the order of token occurrence, the positional vector is added to the input embedding vector:

$$X = E + P, \tag{5}$$

where, P is the positional embedding matrix.

Thanks to this mechanism, the representation of each token is updated based on the context of the entire code segment instead of depending only on neighboring tokens. This is especially useful in vulnerability detection, where the cause of a vulnerability may arise from interactions among statements located far apart. The output of the encoder is the semantic representation matrix shown in Eq. (6) below:

$$H_s = \{h_1, h_2, \ldots, h_L\} \in \mathbb{R}^{L \times d}, \tag{6}$$

where, each vector h_i contains the contextual semantic information of the i-th token.

Step 4: To obtain a fixed representation for the entire code segment, pooling can be applied, for example, averaging over all tokens according to Eq. (7):

$$Z_s = \frac{1}{L}\sum_{i=1}^{L} h_i, \tag{7}$$

where, $Z_s \in \mathbb{R}^d$ is the semantic feature vector of the code segment.

In the SGRL model, the vector Z_s is not used directly for classification but serves as one representation learning branch. In parallel with semantic representation learning using CodeT5, the other branch, also from the normalized source code segments, is represented through a CPG graph and then fed into the GCN network to extract the structural features of the program. The two feature vectors obtained from these two representation learning branches are fused through Concatenation, forming a multi-source representation that integrates both semantic information and structural information. This fused representation serves as the input for the subsequent stages of the model to improve the ability to detect vulnerabilities in source code.

2) Extracting source code features using GCN

a) Overview of GCN: GCN, developed by Thomas Kipf and Max Welling [32], is a variant of Graph Neural Networks (GNNs). A characteristic of GCN is that it uses local spectral filters on graphs to extract graph substructures and thereby reveal graph structure. In terms of its operating principle, the convolution operation in a GCN layer extracts features from neighboring components, similarly to a CNN layer [32, 44]. However, because CNNs do not work well on non-Euclidean datasets, GCN architectures seek to extract attributes from adjacent vertices. GCN is also applied in several fields, such as node classification [45], link prediction [46], graph classification [47], and graph embedding [44]. Eq. (8) below presents the propagation of representative attributes in a GCN layer.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) + H^{(l)} \tag{8}$$

Where:

A: The adjacency matrix of the graph,

X: The input attribute matrix of the node,

$\tilde{D}$: the degree matrix corresponding to $\tilde{A}$,

I: The identity matrix of size N x N,

$\tilde{A} = A + I$,

$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$: The symmetrically normalized combination of the adjacency matrix,

$W^{(l)}$: the learned weight matrix of layer l,

σ: A nonlinear function applied element-wise,

$H^{(l)}$: The output of layer l,

$H^{(0)} = X$.

b) Method for extracting features of the CPG graph based on GCN: In this study, the author selects three main components that describe the structure and information of the CPG graph and also serve as the three inputs to the proposed training model: the adjacency matrix, the node attribute matrix, and the edge attribute matrix. This information is then fed into the GCN model for processing, whose general architecture is shown in Fig. 4.

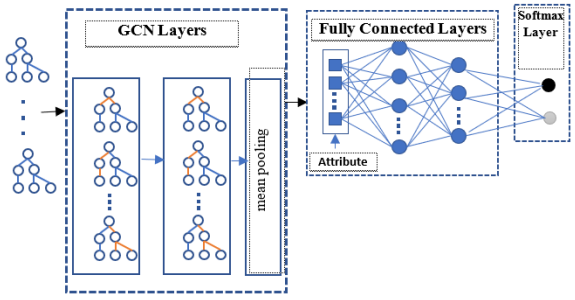


Fig. 4. Architecture of the GCN model.

From Fig. 4, it can be seen that the GCN network architecture performs feature extraction as follows:

- GCN layer: includes GCN layers and a pooling layer to extract graph features. Thus, with the behavior profile as input, the GCN network extracts the characteristics of the graph linking the processes. The output of the GCN layer is hidden representative attributes that show the connections among vertices in the graph through GCN layers. For example, in the case of using two GCN layers, the first GCN layer has the task of considering neighboring vertices of each vertex to update the attributes of that vertex. Next, the second GCN layer recursively updates the attributes of the vertices in the graph. In practice, more than two GCN layers can be used; however, study [44] indicated that only two GCN layers may still ensure a balance between effectiveness and analysis and processing time.
- Fully connected layer: this layer is similar to an MLP network and has the task of learning the attributes processed through the GCN layers.
- Softmax layer: this layer has the task of computing the probabilities of the output labels. The file being predicted is assigned to the class with the highest probability.

3) Building the source code feature vector: After the representation learning stage, the model obtains two complementary feature vectors, including semantic features from CodeT5 and structural features from GCN on CPG. The semantic features reflect the context and relationships among tokens, while the structural features represent syntactic relations, control flow, and data flow. Combining these two feature sources creates a more comprehensive source code representation, effectively supporting the vulnerability detection process.

Assume that $Z_s \in \mathbb{R}^{d_s}$ is the semantic feature vector created by CodeT5 and Z_g is the structural feature vector generated by GCN. The source code feature vector is built by concatenation along the feature dimension as follows:

$$Z = Concat(Z_s, Z_g) = [Z_s; Z_g], \quad (9)$$

Where:

- d_s is the dimension of the semantic feature vector;
- d_g is the dimension of the structural feature vector;
- $Z \in \mathbb{R}^{d_s+d_g}$ is the fused feature vector.

Thus, Concatenation is used to fuse both semantic and structural features of source code. Compared with addition or averaging, Concatenation preserves complete information from the two representation spaces without losing their distinctive features. The fused vector is then fed into subsequent learning layers so that the model can automatically determine the contribution level of each feature type during optimization.

D. Data Rebalancing

After constructing the feature profiles, direct classification may be affected by data imbalance because the numbers of samples in the vulnerable and non-vulnerable classes differ substantially. Traditional methods such as SMOTE and GAN can be used to balance data, but sample generation based on neighborhood relationships in feature space is not entirely suitable for the characteristics of source code data. Therefore, this study proposes a rebalancing mechanism based on BDC to increase the diversity of minority-class representations and reduce the impact of data imbalance during training. Accordingly, for the vector representation, each component becomes: $x = (x_1, x_2, \dots x_m) x_h (h = 1, 2, \dots m)$

$$\widehat{x_h} = a_h \, . \, x_h \quad (10)$$

where, $a_h \sim P$ is the variable is a random variable with a Bernoulli distribution [48]:

$$P(a_h) = \begin{cases} 1-p, & a_h = 0 \\ p \ , & a_h = 1 \end{cases} \quad (11)$$

It can be seen that, in the problem of generating data for the minority class (i.e., the vulnerable class), the Bernoulli distribution will:

- Generate a random value according to the Bernoulli distribution
- Return a result of 0 (non-vulnerable) or 1 (vulnerable)
  - With probability p, it takes the value 1
  - With probability $1 - p$, it takes the value 0

The algorithm below describes the data generation process using BDC.

Algorithm 1: Data balancing using BDC

Input: Feature -$feature$
Label -$label$
Rate -α
Number of generated samples based on minority class -n
$D = \{\}$
Output: $D_{balance}$
Function:$BDC(feature, label, \alpha = 0.1, n = 10)$:
for $f_i, l_i \in feature, label$ do:
if l_i $of\ minority\ class$ do:
for $t := 1$ to n do:
$f'_i \leftarrow generate\_minority\ class\_with\_rate(\alpha)$
$Add\ \ f'_i\ \ in\ D$
end
end
end
return $D_{balance}$
end

From Algorithm 1, it can be seen that:

Step 1: Search for the features of the minority class

Step 2: Iterate through the features of the minority class and apply the Bernoulli generation function to generate data for the minority class.

Step 3: Add the generated data to the original data

E. Source Code Vulnerability Classification

After the representation learning stage, the model obtains a fused feature vector integrating semantic information from CodeT5 and structural information from GCN. This vector is fed into an MLP to classify the source code into two classes: vulnerable and non-vulnerable. MLP learns a nonlinear mapping function from the feature space to the label space, thereby exploiting complex relationships among features and determining the class of the input sample.

Assume that the fused representation vector after the representation learning stage is denoted as:

$$h^* \in \mathbb{R}^d \tag{12}$$

where, d is the feature-space dimension. This input is fed through the hidden layers of the MLP. At the l-th hidden layer, the output is determined according to the equation:

$$h^{(l+1)} = \Phi\left(\mathrm{W}^{(l)} h^{(l)} + b^{(l)}\right), \tag{13}$$

Where:

- $\mathrm{W}^{(l)}$ is the weight matrix of the l-th layer;
- $b^{(l)}$ is the bias vector;
- $\Phi(\cdot)$ is the nonlinear activation function; in this study, the ReLU function is used and defined as:

$$\Phi(x) = \max(0, x) \tag{14}$$

Using multiple hidden layers helps the model learn nonlinear relations between semantic features and structural features, thereby increasing the ability to distinguish vulnerable source code samples from normal source code samples.

After the last hidden layer, the feature vector is mapped to the output layer, whose number of nodes equals the number of classes to be classified. In the source code vulnerability detection problem, the number of output classes is two, corresponding to two labels:

$$y = \{0, 1\} \tag{15}$$

Where:

- 0: the code segment does not contain a vulnerability;
- 1: the code segment contains a vulnerability.

The output logits of the model are computed according to the equation:

$$o = W_o h^{(L)} + b_0 \tag{16}$$

where, W_o are b_0 are the weight and bias of the output layer, respectively.

To convert the logits into predicted probabilities for each class, the model uses the softmax function:

$$P(y = i|x) = \frac{exp(o_i)}{\sum_{j=1}^{2} exp(o_j)}, i\epsilon\{0, 1\}, \tag{17}$$

Where:

- o_i: is the logit of the i-th class;
- $P(y = i|x)$: is the probability that the code segment belongs to class i.

The final predicted label is determined according to the maximum-probability criterion:

$$\hat{y} = \arg\max_i P(y = i|x) \tag{18}$$

During training, the parameters of the entire model are optimized by minimizing the Cross-Entropy loss function, defined as follows:

$$\mathrm{L} = -\sum_{i=1}^{2} y_i \,.\, log\ P(y = i|x) \tag{19}$$

Where:

- y_i: is the actual label of the i-th class;
- $P(y = i|x)$: is the probability predicted by the model.

The optimization process is performed by the Adam algorithm, in which the model parameters are updated in the direction of decreasing the value of the loss function on the training dataset. Thanks to its effective optimization capability for deep learning models, Adam helps speed up convergence and improve stability during training.

Through the combination of semantic representation from CodeT5, structural representation from GCN, and the MLP classifier, the model can learn distinguishing features of vulnerable samples even when they appear in many different forms in source code. At the same time, using an MLP allows the model to effectively exploit nonlinear interactions among the fused features, contributing to improving the accuracy of the source code vulnerability detection problem.

IV. Experiments and Evaluation

A. Experimental Dataset

Table I presents statistics on the experimental datasets proposed for use in the study.

TABLE I. Experimental Data Statistics

	Total number of samples	Number of vulnerabilities	Number of normal samples
Verum Dataset [41]	18169	1658	16511
FFmpeg+Qume Dataset [49]	25905	11804	14101

The Verum dataset [41] is built from open-source projects with CVE-coded vulnerabilities [50], in which code regions

containing vulnerabilities are labeled as vulnerable and safe regions are labeled as non-vulnerable. Verum supports research on vulnerability detection in C/C++ source code. Study [41] shows that this dataset has a large imbalance between security vulnerability data and normal data, with a small code duplication rate of about 0.6%, and is therefore suitable for the code vulnerability classification task.

To demonstrate the effectiveness of the proposed model, the study evaluates it using the two main datasets summarized in Table I.

B. Experimental Scenarios and Evaluation Criteria

1) Experimental scenarios: To demonstrate the effectiveness of the proposed model, the author conducts the following experimental scenarios:

a) Scenario 1: How effective is the author's proposed method in classifying source code vulnerabilities?

b) Scenario 2: Evaluation of the role of the individual components in the author's proposed model. Through this, it is confirmed whether the author's proposed method is more effective than the separate approach of each component in the proposed model on the same experimental dataset.

c) Scenario 3: Comparing the SGRL model with several other approaches on the same experimental dataset.

2) Evaluation criteria: To evaluate effectiveness, the study uses several metrics:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (20)$$

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (21)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (22)$$

$$F1 - score = \frac{2 \times precision \times Recall}{precision + Recall} \times 100\% \quad (23)$$

Where:

- Accuracy: The proportion of correctly classified samples over the total number of samples. Accuracy is calculated according to (20).
- Precision: The proportion of TP over (TP + FP), reflecting precision when predicting positive (21).
- Recall: The proportion of TP over (TP + FN), indicating the ability to correctly detect positive samples (22).
- F1-score: The harmonic mean of Precision and Recall; the higher it is, the better the classifier (23).

C. Experimental Result on the Proposed Model

1) Experimental results of Scenario 1: In this experimental scenario, the author conducts experiments on the Verum dataset. Through the dataset description in Section IV, Part A, study [41] also confirmed that this dataset is very suitable for application to security vulnerability classification tasks. Table II below presents statistics on the effectiveness of the proposed method in security vulnerability detection based on several criteria defined in Section IV-B-2.

TABLE II. EXPERIMENTAL RESULTS OF SCENARIO 1

BDC	GCN_CodeT5											
	128				256				512			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
0.1	81.85	81.95	81.85	81.79	82.59	82.67	82.59	82.55	83.48	83.81	83.48	83.39
0.2	82.59	82.74	82.59	82.53	83.04	83.18	83.04	82.98	**84.23**	**84.55**	**84.23**	**84.14**
0.3	82.29	82.41	82.29	82.24	82.89	83.04	82.89	82.83	84.08	84.34	84.08	84.0

From the experimental results of Scenario 1 in Table II, the model combining GCN and CodeT5 with an input length of 512 tokens and a BDC probability of 0.2 is the most suitable configuration among the parameters investigated. This configuration achieves an accuracy of 84.23%, Precision of 84.55%, Recall of 84.23%, and F1-score of 84.14%, attaining the highest values for all four evaluation metrics. The results indicate that increasing the token sequence length to 512 enables the model to exploit the semantic context of source code more effectively, while a BDC probability of 0.2 provides an appropriate degree of regularization for representation learning. Increasing the BDC probability further to 0.3 leads to a slight decline in performance, indicating that excessive feature removal may discard useful information. Therefore, combining a token length of 512 with BDC = 0.2 is the optimal configuration, yielding the best and most consistent results across all the evaluation metrics of the proposed model. This is also scientifically reasonable because, according to the survey presented in this study and the study by Chakraborty et al. [41], as many as 70% of the source code files in the Verum dataset have lengths greater than 256 and less than 512. Accordingly, selecting a token length of 512 yields the best model performance.

Fig. 5 presents the confusion matrix calculated for the proposed model with a token length of 512 and a BDC probability of 0.2.

Thus, although the Verum dataset exhibits severe class imbalance, with secure source code accounting for approximately 90.88%, the SGRL model under the 512/0.2 configuration demonstrates superior identification capability by correctly detecting 1,397 of the 1,658 actual vulnerable samples (Recall = 84.23%). Meanwhile, the number of false positives (FP) remains very low (255 samples out of a total of 16,511 secure samples), maintaining Precision at 84.55% and minimizing the cost of manual code inspection in practice.

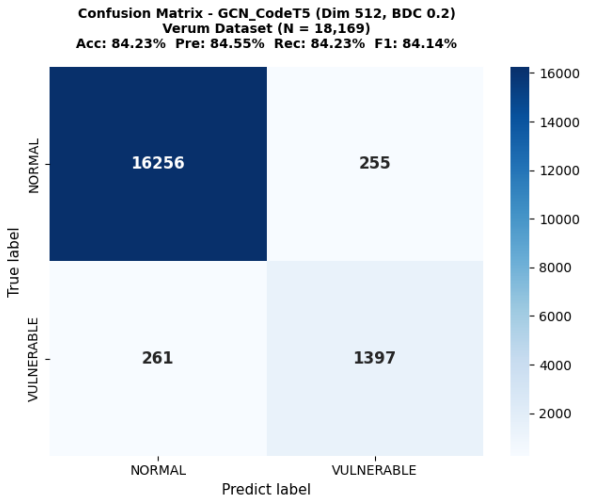


Fig. 5. Confusion matrix values of the proposed SGRL model.

2) Experimental results of Scenario 2: In this scenario, to demonstrate the effectiveness of the proposed model, the author conducts experiments on each component of the proposed model to prove that the combination of the GCN deep graph network with CodeT5 gives the best result.

First, the author conducts an experiment using only CodeT5 to extract source code features, combined with BDC to rebalance the vulnerable and non-vulnerable labels in the experimental dataset. The resulting features are then passed through a Multi-Layer Perceptron (MLP) feedforward neural network to classify source code as vulnerable or non-vulnerable.

Table III shows that, when the model is evaluated using individual components—in this case, using only CodeT5 for source code feature extraction—the resulting architecture performs considerably worse than the model proposed in this study. Specifically, at the same configuration of a 512-token input length and BDC probability of 0.2, which yields the best results among the investigated token lengths of 128, 256, and 512 and their corresponding BDC probabilities, the Accuracy, Precision, Recall, and F1-score are lower than those of the proposed model by 0.9%, 1.11%, 0.9%, and 0.85%, respectively.

TABLE III. EXPERIMENTAL RESULTS WHEN USING ONLY THE CODET5 MODEL FOR SOURCE CODE FEATURE EXTRACTION

BDC	CodeT5											
	128				256				512			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
0.1	81.55	81.56	81.55	81.53	82.14	82.15	82.14	82.15	82.44	82.49	82.44	82.41
0.2	81.55	81.56	81.55	81.53	82.14	82.15	82.14	82.15	**83.33**	**83.44**	**83.33**	**83.29**
0.3	81.55	81.56	81.55	81.53	82.14	82.15	82.14	82.15	82.44	82.49	82.44	82.41

In the second case, the same configuration is retained, but the source code feature extraction method is changed by replacing the combined GCN and CodeT5 components with the CodeBERT model. Experiments using token lengths of 128, 256, and 512 and corresponding BDC probabilities of 0.1, 0.2, and 0.3 yield the results shown in Table IV below:

TABLE IV. EXPERIMENTAL RESULTS WHEN USING ONLY THE CODEBERT MODEL FOR SOURCE CODE FEATURE EXTRACTION

BDC	CodeBERT											
	128				256				512			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
0.1	82.59	83.04	82.59	82.47	83.93	84.25	83.93	83.84	82.44	82.91	82.44	82.31
0.2	82.74	83.37	82.74	82.58	82.44	82.82	82.44	82.33	**92.59**	**82.91**	**82.95**	**82.49**
0.3	82.44	82.78	82.44	82.34	82.74	82.93	82.74	82.67	82.59	82.6	82.59	82.59

Table IV indicates that the best result is obtained with a token length of 512 and a corresponding BDC probability of 0.2. However, when GCN combined with CodeT5 is used for source code feature extraction, the results are considerably better in terms of Precision, Recall, and F1-score, by 1.64%, 1.28%, and 1.65%, respectively. Only the Accuracy of CodeBERT when used independently is approximately 8.36% higher than that of the author's proposed GCN–CodeT5 model. However, higher Accuracy alone is insufficient to establish that a model has better vulnerability detection capability. For vulnerability detection, the ability to correctly identify vulnerable samples, as reflected by Recall, is particularly important. The proposed model combining GCN and CodeT5 achieves Recall of 84.23% and F1-score of 84.14%, which are higher than those of CodeBERT by 1.28% and 1.65%, respectively, indicating a better balance between detecting vulnerable samples and limiting incorrect predictions. Thus, although CodeBERT achieves higher Accuracy, the model combining GCN and CodeT5 demonstrates a more balanced performance and is better suited to the objective of source code vulnerability detection.

In the third case, only the GraphCodeBERT model is used to replace the GCN and CodeT5 components in the author's proposed model. With experimental token lengths of 128, 256, and 512 and corresponding BDC probabilities of 0.1, 0.2, and 0.3, the results shown in Table V below are obtained:

TABLE V. EXPERIMENTAL RESULTS WHEN USING ONLY THE GRAPHCODEBERT MODEL FOR SOURCE CODE FEATURE EXTRACTION

BDC	GraphCodeBERT											
	128				256				512			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
0.1	83.78	84.16	83.78	83.68	82.14	82.76	82.14	81.98	83.63	83.81	83.63	83.57
0.2	83.78	84.16	83.78	83.68	82.14	82.76	82.14	81.98	**83.48**	**83.67**	**83.48**	**83.42**
0.3	83.78	84.16	83.78	83.68	82.14	82.76	82.14	81.98	83.48	83.7	83.48	83.41

From Table V, it can be clearly seen that the replacement model GraphCodeBERT gives poorer classification results than the model proposed in the study. Specifically, the accuracy, precision, recall, and F1-score are lower than those of the proposed model by 0.75%, 0.88%, 0.75%, and 0.72%, respectively. This result demonstrates the superiority of the combined CodeT5 and GCN model for the source code feature extraction task, proving the correctness and reasonableness of the authors' proposed model.

3) Experimental results of Scenario 3: Comparison of the SGRL model with several other approaches.

This scenario aims to evaluate and compare the SGRL model with several other studies on different experimental datasets. The two datasets used are Verum and FFmpeg+Qume, representative datasets currently widely employed in source code vulnerability detection. The Verum dataset is used in Scenarios 1 and 2 of this study, whereas FFmpeg+Qume is used in Scenario 3 to demonstrate that the SGRL model performs well not only on one dataset but also on different datasets. The experimental results for Scenario 3 are presented in Table VI.

The experimental results in Table VI show that the author's proposed method achieves substantially better results on most metrics. In this study, the author reuses the experimental results reported in study [41] to compare the proposed approach with REVEAL [41], Russell [51], VulDeePecker [52], SySeVR [55], and Devign [56]. Comparing the experimental results in Table VI with those in Table II for the Verum dataset also indicates that the proposed method performs substantially better than the other approaches. Specifically, compared with REVEAL [41], the author's proposal achieves higher Precision, Recall, and F1-score by 50.52%, 19.99%, and 39.65%, respectively; only its Accuracy is 2.71% lower than that of REVEAL. Similarly, the results of Russell [51], VulDeePecker [52], SySeVR [55], and Devign [56] are lower than those of the author's method across all metrics, ranging from a minimum difference of 66.87% in Precision to a maximum difference of 70.36% in Recall. On the FFmpeg+Qume dataset [49], the author's proposal achieves substantially better results on all metrics than the other methods, including Russell [51], VulDeePecker [52], SySeVR [55], and Devign [56], evaluated on the same FFmpeg+Qume experimental dataset. Specifically, the maximum relative improvements of the author's proposal over the other studies in Accuracy, Precision, Recall, and F1-score are 38.3%, 42.29%, 63.14%, and 55.53%, respectively. This is a very good result, indicating that the model proposed in this study is effective not only on one experimental dataset but also on multiple datasets. Its effectiveness is reflected in the experimental results, in which the proposed model achieves good results that exceed those of all other approaches on most metrics.

TABLE VI. COMPARISON RESULTS FOR THE SGRL MODEL AND OTHER APPROACHES ON THE VERUM AND FFMPEG+QUME DATASETS

Experimental Dataset	Approach	Accuracy	Precision	Recall	F1-score
Verum	**Article proposal**	**84.23**	**84.55**	**84.23**	**84.14**
	REVEAL [41]	86.94	34.03	64.24	44.49
	Russell [51]	90.98	24.63	10.91	15.24
	VulDeePecker [52]	89.05	17.68	13.87	15.7
	SySeVR [53]	84.22	24.46	40.11	30.25
	Devign [54]	88.41	34.61	26.67	29.87
FFmpeg+Qume	**Article proposal**	**90.82**	**89.65**	**91.84**	**90.73**
	REVEAL [41]	62.51	56.85	74.61	64.42
	Russell [51]	58.13	54.04	39.50	45.62
	VulDeePecker [52]	53.58	47.36	28.70	35.20
	SySeVR [53]	52.52	48.34	65.96	56.03
	Devign [54]	58.57	53.60	62.73	57.18

Fig. 6 presents the confusion matrix calculated for the proposed model with a token length of 512 and a BDC probability of 0.2.

For the FFmpeg+Qume dataset, which has a more balanced label distribution, the SGRL model demonstrates stability and strong generalization capability by correctly predicting 9,943 vulnerable samples and 12,285 normal samples. The false-

negative (FN) and false-positive (FP) counts remain symmetrically balanced across the main diagonal, demonstrating that combining graph structure (GCN) with the CodeT5 source code semantic representation model enables the system to process source code functions of substantial length and complexity effectively.

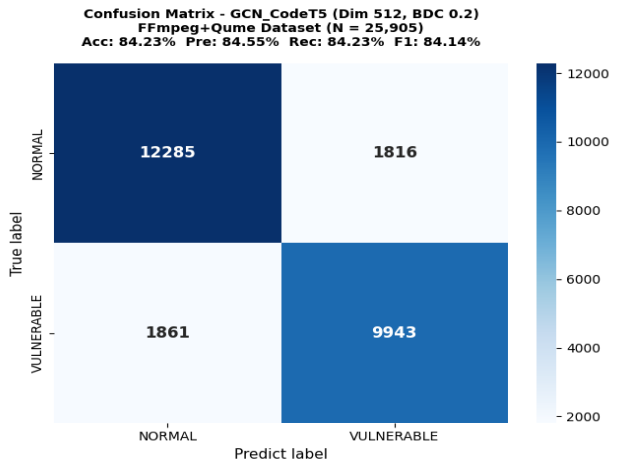


Fig. 6. Confusion matrix values of the SGRL model on the FFmpeg+Qume dataset.

V. Conclusion

Finally, from an academic perspective, the study contributes to expanding the research direction of multi-source representation learning for source code vulnerability detection by integrating a pre-trained language model with a graph neural network in a unified architecture. The research results not only affirm the complementary role between semantic representation and structural representation, but also provide an experimental basis for developing next-generation vulnerability detection models capable of simultaneously exploiting multiple sources of information from source code. This is also a premise for extending the SGRL Framework toward combining Graph Transformer, Graph Attention Network, or adaptive feature fusion mechanisms in subsequent studies. Although the proposed method has achieved promising results, the study still has several limitations. First, the method depends on the ability of Joern to analyze source code and construct the Code Property Graph (CPG). For source code with complex structures, specific syntactic characteristics, or features that are not fully supported by the tool, the quality of the generated CPG may be affected, thereby influencing the effectiveness of structural feature representation. Second, the proposed architecture integrates both Transformer and GCN branches to extract semantic and structural features, respectively, and may therefore require greater computational and memory resources than methods employing a single representation branch. These limitations should be considered when deploying the proposed method in real-world software development environments. In future research, the authors will focus on improving the ability to process complex source code and optimizing the architecture to reduce computational costs while maintaining the capability to simultaneously exploit semantic and structural information from source code. These efforts will also provide a foundation for extending the SGRL framework by integrating Graph Transformer models, Graph Attention Networks, or adaptive feature fusion mechanisms in subsequent studies.

References

[1] G. Lin, S. Wen, Q.-L. Han, J. Zhang and Y. Xiang, "Software Vulnerability Detection Using Deep Neural Networks: A Survey," *in Proceedings of the IEEE,* vol. 108, no. 10, pp. 1825-1848, Oct. 2020, doi: 10.1109/JPROC.2020.2993293.

[2] "The National Institute of Standards and Technology (NIST)," 2024. [Online]. Available: https://www.nist.gov/. [Accessed 16 9 2026].

[3] Z. Liu, L. Pan and C. Chen et al., "VulMatch: Binary-level Vulnerability Detection Through Signature," in *Network and System Security: 18th International Conference, NSS 2024, Abu Dhabi, United Arab Emirates, November 20–22, 2024, Proceedings*, 2024, pp. 409-427, https://doi.org/10.1007/978-981-96-3531-3_20.

[4] X. Li, L. Wang, Y. Xin, Y. Yang, Q. Tang and Y. Chen, "Automated Software Vulnerability Detection Based on Hybrid Neural Network," *Applied Sciences,* vol. 11, no. 7, p. 3201, 2021, https://doi.org/10.3390/app11073201.

[5] H.-H. Wei and M. Li, "Supervised deep features for software functional clone detection by exploiting lexical and syntactical information in source code," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, August 2017. DOI: 10.24963/ijcai.2017/423.

[6] J. DevlinMing, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT 2019,* p. 4171–4186, October 2018, DOI: 10.48550/arXiv.1810.04805.

[7] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," in *International Conference on Learning Representations*, 2019, https://arxiv.org/abs/1907.11692.

[8] Z. Feng, D. Guo and D. Tang et al., "CodeBERT: A Pre-Trained Model for Programming and Natural Languages," in *Conference: Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, DOI: 10.18653/v1/2020.findings-emnlp.139.

[9] Y. Wang, W. Wang, and S. Joty et al., "CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation," in *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, 2021, pp. 8696-8708, DOI: 10.18653/v1/2021.emnlp-main.685.

[10] F. Yamaguchi, M. Lottmann and K. Rieck, "Generalized vulnerability extrapolation using abstract syntax trees," in *ACSAC '12: Proceedings of the 28th Annual Computer Security Applications Conference*, vol. 28, pp. 359-368, 2012, DOI: 10.1145/2420950.2421003.

[11] H. Gascon et al., "Structural detection of android malware using embedded call graphs," in *ACM workshop on Artificial intelligence and security*, New York, USA, pp. 45-54, 2013, https://doi.org/10.1145/2517312.2517315.

[12] J. Ferrante, K. J. Ottenstein and J. D. Warren, "The program dependence graph and its use in optimization," *ACM Transactions on Programming Languages and Systems,* vol. 9, no. 3, pp. 319-349, 1987.

[13] F. Yamaguchi, N. Golde, D. Arp and Konrad Rieck, "Modeling and Discovering Vulnerabilities with Code Property Graphs," in *2014 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 18-21 May 2014, doi: 10.1109/SP.2014.44.

[14] A. Abiri, A. Tajary and M. Fateh, "Impact of Oversampling Methods on Imbalanced Dataset for Software Fault Prediction," in *2025 15th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2025, DOI: 10.1109/ICCKE68588.2025.11273874.

[15] Y. Guo, Q. Hu and Q. Tang et al., "An Empirical Study of the Imbalance Issue in Software Vulnerability Detection," in *Computer Security – ESORICS 2023: 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25–29, 2023, Proceedings, Part IV*, 2024, pp 371–390, vol. 14347, https://doi.org/10.1007/978-3-031-51482-1_19.

[16] S. S. Daneshvar, D. Tan and S. Wang et al., "A Study on Mixup-Inspired Augmentation Methods for Software Vulnerability Detection," in *EASE '25: Proceedings of the 29th International Conference on Evaluation and*

Assessment in Software Engineering, 2025, pp. 616-621, https://doi.org/10.1145/3756681.3757017.

[17] P. Haridas, G. Chennupati and N. Santhi et al., "Code Characterization With Graph Convolutions and Capsule Networks," *in IEEE Access,* vol. 8, pp. 136307-136315, 2020, doi:10.1109/ACCESS.2020.3011909..

[18] B. Wu, F. Zou and P. Yi et al., "SlicedLocator: Code vulnerability locator based on sliced dependence graph," *Computers & Security,* vol. 134, no. C, p. 103469, 2023, https://doi.org/10.1016/j.cose.2023.103469.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res,* vol. 16, no. 1, p. 321–357, 2002.

[20] I. J. Goodfellow, J. Pouget-Abadie and M. Mirza et al., "Generative Adversarial Nets," *in Advances in Neural Information Processing Systems (NeurIPS),* vol. 27, p. 2672–2680, 2014.

[21] D. Chen, Y.-d. Zhang and . W. Wei et al., "Efficient vulnerability detection based on an optimized rule-checking static analysis technique," *Frontiers of Information Technology & Electronic Engineering,* vol. 18, pp. pp. 332-345, 2017.

[22] R. Croft, D. Newlands and Z. Chen et al., "An Empirical Study of Rule-Based and Learning-Based Approaches for Static Application Security Testing," in *ESEM '21: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1-12, No. 8, 2021, https://doi.org/10.1145/3475716.3475781.

[23] "CWE TOP25," 2024. [Online]. Available: https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html. [Accessed 16 9 2026].

[24] S. H. Ding, B. C. Fung and P. Charland, "Asm2Vec: Boosting Static Representation Robustness for Binary Clone Search against Code Obfuscation and Compiler Optimization," in *in Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 19-23 May 2019, IEEE, pp. 472–489, DOI: 10.1109/SP.2019.00003.

[25] X. Li, Y. Qu and H. Yin, "Palmtree: Learning an assembly language model for instruction embedding," in *in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3236–3251, https://doi.org/10.1145/3460120.3484587.

[26] T. Mikolov, K. Chen and G. Corrado et al., "Efficient Estimation of Word Representations in Vector Space," 2013, arXiv:1301.3781.

[27] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning, PMLR*, vol. 32, no. 2, pp. 1188-1196, 2014.

[28] Y. Ding, Y. Fu and O. Ibrahim et al., "Vulnerability Detection with Code Language Models: How Far Are We?," in *ICSE '25: Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, pp.1729-1741, 2025, https://doi.org/10.1109/ICSE55347.2025.00038.

[29] I. Kalouptsoglou, M. Siavvas and A. Ampatzoglou et al., "Transfer learning for software vulnerability prediction using Transformer models," *Journal of Systems and Software,* vol. 227, p. 112448, 2025, https://doi.org/10.1016/j.jss.2025.112448.

[30] G. Zhang, T. Yao and J. Qin et al., "CodeSAGE: A multi-feature fusion vulnerability detection approach using code attribute graphs and attention mechanisms," *Journal of Information Security and Applications,* vol. 89, p. 103973, 2025, https://doi.org/10.1016/j.jisa.2025.103973.

[31] R. Liu, Y. Wang and H. Xu et al., "Vul-LMGNNs: Fusing Language Models and Online-Distilled Graph Neural Networks for Code Vulnerability Detection," *Information Fusion,* vol. 115, p. 102748, March 2025. https://doi.org/10.1016/j.inffus.2024.102748.

[32] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *International Conference on Learning Representations 2017 (ICLR 2017)*, 2017.

[33] D. Hin, A. Kan and H. Chen et al., "Linevd: Statement-level vulnerability detection using graph neural networks," in *In Proceedings of the 19th international conference on mining software repositories*, pp. 596–607, 2022.

[34] H. Wang et al., "Combining Graph-Based Learning With Automated Data Collection for Code Vulnerability Detection," *in IEEE Transactions on Information Forensics and Security,* vol. 16, pp. 1943-1958, 2021, doi: 10.1109/TIFS.2020.3044773..

[35] J. Chen, Y. Yin and S. Cai et al., "iGnnVD: A novel software vulnerability detection model based on integrated graph neural networks," *Science of Computer Programming,* vol. 238, no. C, p. 103156, 2024, https://doi.org/10.1016/j.scico.2024.103156.

[36] C. Liang, Q. Wei and Z. Jiang et al., "A Source Code Vulnerability Detection Method Based on Adaptive Graph Neural Networks," in *ASEW '24: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 2024, pp. 187-196, https://doi.org/10.1145/3691621.3694950.

[37] Y. Luo, W. Xu and D. Xu , "Detecting code vulnerabilities with heterogeneous GNN training," *International Journal of Information Security,* vol. 24, no. 213, 2025, https://doi.org/10.1007/s10207-025-01132-x.

[38] X. Cao, J. Wang and P. Wu et al., "VulMPFF: A Vulnerability Detection Method for Fusing Code Features in Multiple Perspectives," *IET Information Security,* vol. 1, pp. pp. 1-15, 2024, https://doi.org/10.1049/2024/4313185.

[39] B. V. Cong and D. X. Cho, "Detecting software vulnerabilities based on source code analysis using GCN Transformer," in *in 2023 RIVF Int. Conf. Comput*, Hanoi, Vietnam, 2023.

[40] C. V. Bui, C. X. Do and T. T. Do, "Detection of source code vulnerabilities using Nature language processing and deep graph network," *Journal of Science and Technology on Information security,* vol. 23, no. 3, pp. 27-42, 2024, DOI: 10.54654/isj.v3i23.1057.

[41] S. Chakraborty, R. Krishna and Y. Ding et al., "Deep Learning based Vulnerability Detection: Are We There Yet?," *IEEE Transactions on Software Engineering,* vol. 48, pp. 3280-3296, Sept. 2022, DOI: 10.1109/TSE.2021.3087402.

[42] "JOERN, The Bug Hunter's Workbench," 2024. [Online]. Available: https://joern.io/. [Accessed 16 9 2026].

[43] B. V. Cong and C. D. Xuan, "A New Framework for Software Vulnerability Detection Based on an Advanced Computing," *Computers, Materials & Continua,* vol. 79, no. 3, pp. 3700-3723, 2024, DOI: 10.32604/cmc.2024.050019.

[44] J. Busch et al., "NF-GNN: Network Flow Graph Neural Networks for Malware Detection and Classification," in *SSDBM '21: Proceedings of the 33rd International Conference on Scientific and Statistical Database Management*, pp. 121 - 132, 2021.

[45] M. Chen, Z. Wei and Z. Huang et al., "Simple and Deep Graph Convolutional Networks," in *Proceedings of the 37th International Conference on Machine Learning, PMLR*, Vol. 119, pp. 1725-1735, 2020.

[46] K. KC, R. Li, F. Cui and A. Haake, "Predicting Biomedical Interactions with Higher-Order Graph Convolutional Networks," *Journal of latex class files,* vol. 14, no. 8, p. 99, 2020.

[47] M. Balcilar and G. Renton et al., "Bridging the Gap Between Spectral and Spatial Domains in Graph Neural Networks," 2020.

[48] J. K. Blitzstein et al., Introduction to Probability, Boca Raton, FL 33487-2742, USA: Taylor & Francis Group, LLC, ISBN-13: 978-1-1383-6991-7, 2019.

[49] "Download FFmpeg," 2026. [Online]. Available: https://ffmpeg.org/download.html. [Accessed 16 9 2026].

[50] "CVE," 2024. [Online]. Available: http://cve.mitre.org. [Accessed 16 9 2026].

[51] G. Siewruk and W. Mazurczyk, "Context-Aware Software Vulnerability Classification Using Machine Learning," *in IEEE Access,* vol. 9, pp. 88852-88867, 2021, doi: 10.1109/ACCESS.2021.3075385.

[52] Z. Li et al., "VulDeePecker: A deep learning-based system for vulnerability detection," in *Network and Distributed Systems Security (NDSS) Symposium 2018, 18-21 February 2018,*, San Diego, CA, USA, 2018, http://dx.doi.org/10.14722/ndss.2018.23158..

[53] Z. Li et al., "SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities," *in IEEE Transactions on Dependable and Secure Computing,* vol. 19, pp. 2244-2258, doi: 10.1109/TDSC.2021.3051525., 2022.

[54] Y. Zhou, S. Liu and J. Siow et al., "Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks," *In: Advances in Neural Information Processing Systems,* no. 915, p. 10197–10207, 2019.

[55] Z. Li et al., "SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities," *in IEEE Transactions on Dependable and Secure Computing,* vol. 19, pp. 2244-2258, 2022, doi: 10.1109/TDSC.2021.3051525..

[56] Y. Zhou, S. Liu and J. Siow et al., "Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks," *In: Advances in Neural Information Processing Systems,* no. 915, p. 10197–10207, 2019, https://doi.org/10.48550/arXiv.1909.03496.

Cross-Cohort Reliability of Machine Learning for Breast Cancer Survival Prediction

NADIR Younes[1]*, RACHDI Mohamed[2], AMHAIMAR Lahcen[3],
BAKHOUYI Abdellah[4], KHALIDI Abderrahim[5], AZOUAZI Mohamed[6]
M2S2I Laboratory-National Higher School of Art and Design (ENSAD),
Hassan II University of Casablanca, Casablanca, Morocco[1, 2, 3, 4, 5]
LTIM / Laboratory of Information Technology and Modeling-Faculty of Sciences Ben M'Sik (FSBM),
Hassan II University of Casablanca, Casablanca, Morocco[6]

***Abstract*—Transportability limitations can affect discrimination, probability calibration, and uncertainty estimates even when machine-learning pipelines use the same input variables. This study evaluated 5-year breast cancer survival models across METABRIC and TCGA-BRCA using age and ER, PR, and HER2 status. Logistic regression, radial-basis-function support vector machine, random forest, XGBoost, and soft voting underwent bidirectional external validation. The METABRIC soft-voting model was further assessed with source-only Platt scaling, split conformal prediction, selective prediction by singleton sets, cohort classification, and exploratory few-shot target adaptation. The principal contribution is an integrated reliability framework. METABRIC-to-TCGA ROC-AUC decreased numerically from 0.732 to 0.643, an observed difference of approximately 0.088 with a 95 percent confidence interval from minus 0.015 to 0.196. TCGA-to-METABRIC ROC-AUC decreased numerically from 0.674 to 0.586, also by approximately 0.088, with a 95 percent confidence interval from minus 0.127 to 0.282. Neither interval excluded zero. Source Platt scaling improved the observed internal Brier score and expected calibration error but produced worse observed values for both metrics externally. TCGA empirical conformal coverage was 81.8 percent versus nominal 90 percent. Exploratory target recalibration improved mean locked-test Brier score and expected calibration error at larger target sample sizes, whereas updating only the conformal threshold did not restore nominal mean coverage. These findings support separate evaluation of ranking, probability accuracy, and uncertainty under cohort shift and do not establish model superiority, clinical usefulness, or deployment readiness.**

Keywords*—Machine learning; transportability; external validation; probability calibration; conformal prediction; selective prediction; cohort shift; few-shot recalibration; survival prediction

I. Introduction

Five-year overall survival is a clinically interpretable endpoint, and breast cancer contributes substantially to global cancer incidence [1]. Prognosis and the data used to estimate it vary across patient populations and data-generating settings. Machine-learning models can combine clinical predictors through linear, kernel-based, tree-based, and ensemble relationships, yet apparent performance within a development cohort does not establish performance in a distinct population [2]. Differences in case mix, variable definitions, missingness, care pathways, and follow-up can alter discrimination and probability calibration even when the same predictor names are available.

External validation is therefore central to evaluating model transportability [3,4]. Discrimination measures whether a model ranks patients with different outcomes; calibration measures agreement between predicted probabilities and observed outcome frequencies [4,5]. A transported model may retain ranking but yield inaccurate probabilities. A calibration map fitted in the source cohort may not improve calibration in the target cohort.

Uncertainty methods introduce a second transportability question. Split conformal classification can produce patient-specific prediction sets with finite-sample marginal coverage under exchangeability [6]. That guarantee does not automatically extend to a target cohort under distribution shift [7]. Empirical coverage and abstention behavior must consequently be evaluated in the target cohort rather than assumed. Limited labeled target data may permit probability recalibration or re-estimation of a conformal threshold, but small adaptation samples can introduce instability.

These issues were investigated in two clinical breast cancer cohorts, METABRIC and the TCGA-BRCA publication cohort. The study had four objectives: quantify observed internal-to-external performance changes in both transport directions; assess whether source-only Platt scaling transported from METABRIC to TCGA; evaluate source split-conformal coverage and conformal abstention internally and externally; and explore whether few-shot target adaptation improved probability calibration or restored nominal conformal coverage on a locked TCGA test set. The study design and protected information boundaries are summarized in Fig. 1 in Section III.

The principal contribution of this work is an integrated cross-cohort machine-learning reliability framework. It examines discrimination transportability, probability calibration, conformal coverage, selective prediction, cohort shift, and limited-label target adaptation under protected external evaluation.

Specifically, the study contributes: 1) bidirectional METABRIC↔TCGA external validation of five machine-learning models; 2) joint assessment of discrimination, probability calibration, and conformal uncertainty rather than reliance on accuracy or ROC-AUC alone; 3) evaluation of

*Corresponding author.

whether source-derived Platt calibration and empirical conformal coverage remain reliable after cohort transport; and 4) exploratory few-shot target adaptation with a permanently locked target test, separating probability recalibration from conformal-threshold adaptation.

The remainder of this study is organized as follows. Section II reviews related work on external validation, calibration, dataset shift, and conformal prediction. Section III describes the cohorts, endpoint, protected modeling workflow, calibration and conformal procedures, and statistical analysis. Section IV presents cross-cohort performance, calibration, uncertainty, domain-shift, and exploratory few-shot results. Section V discusses the findings and outlines future work; Section VI summarizes limitations, and Section VII concludes the study.

II. Related Work

METABRIC and TCGA-BRCA have both supported research on breast cancer biology and outcomes, but they differ in study design, composition, and available clinical fields [8,9]. Similar variable names cannot be assumed to have identical meanings across the two studies. A systematic review of 31 machine-learning studies of 5-year breast-cancer survival found that all used internal validation, but only one used external validation and only one assessed calibration [2]. Later work externally evaluated 87 breast-cancer prediction models [10] and compared statistical and machine-learning approaches through internal-external validation [11]. These studies support assessing discrimination and calibration in new populations. Related image-based work combined CLAHE-enhanced mammograms with EfficientNetV2S for breast-cancer classification [12]. A systematic review also examined deep-learning methods for breast-cancer classification [13]. These image-based diagnostic tasks differ from transporting clinical survival probabilities across cohorts.

Clinical prediction guidance recommends protected separation of development, tuning, calibration, and final evaluation, together with uncertainty-aware reporting [3,4,14]. Dataset shift can produce performance changes after deployment. Recent clinical work studied stable prediction under temporal electronic-health-record shift [15]. Cohort-membership classifiers provide a complementary label-free measure of multivariable shift [16]. Covariate differences alone do not establish that transported probabilities or uncertainty sets remain reliable.

Calibration can change even when ranking is retained [5]. Platt scaling uses a parsimonious logistic score transformation [17]. Model-updating research includes slope-and-intercept recalibration with attention to sample size [18], broader guidance on recalibration and revision [19], target-time correction of calibration shift [20], and dynamic updating under changing clinical environments [21]. This literature supports evaluating limited-label updating. It does not justify using the target test set to choose an updating method or sample size.

Conformal prediction provides set-valued uncertainty under exchangeability, and weighted approaches address specified covariate-shift settings [6,7]. More recent work extends conformal inference beyond exchangeability [22] and develops doubly robust calibration of prediction sets under covariate shift [23]. Selective-prediction methods separately formalize abstention or rejection [24,25]. The present study evaluates conventional source split-conformal sets and singleton-based selective prediction empirically after transport; it does not claim the guarantees of these shift-robust extensions.

Existing studies address different parts of this reliability problem. Breast-cancer external-validation studies commonly assess discrimination and calibration, whereas methodological work examines shift, conformal prediction, selective prediction, or model updating [2,10,11,19,22,23,25]. This study evaluates those components in one protected patient-level cross-cohort workflow.

III. Materials and Methods

A. Study Design and Cohorts

This was a retrospective secondary analysis of clinical data from METABRIC (brca_metabric) and TCGA-BRCA (brca_tcga_pub). Only patient-level and sample-level clinical data obtained through the official cBioPortal REST API were used; no mutation, expression, copy-number, imaging, or complete study-archive data were included. The unit of analysis was the patient. Stable patient identifiers were required and checked for duplication before modeling.

Each cohort served once as the source and once as the external cohort. The bidirectional baseline analyses transported METABRIC models to all eligible TCGA patients and TCGA models to all eligible METABRIC patients. Source calibration, split-conformal, and domain-shift analyses used the fixed METABRIC soft-voting model with TCGA as the external cohort. The few-shot analysis was exploratory and post hoc. Fig. 1 shows the protected workflows. Panel A separates source training, disjoint probability and conformal calibration, and evaluation without refitting; it also shows the complementary TCGA-to-METABRIC baseline direction. Panel B shows separate target probability and conformal-threshold branches, each evaluated only on the locked TCGA test.

B. Five-Year Overall-Survival Outcome

Overall survival status was harmonized from OS_STATUS, with 0: LIVING mapped to living and 1: DECEASED mapped to deceased. Overall survival time was parsed from nonnegative OS_MONTHS. The binary endpoint used a 60-month horizon. Death on or before 60 months was assigned y_5yr=0; documented survival beyond 60 months was assigned y_5yr=1, irrespective of a later death or censoring event. Patients recorded as living or censored on or before 60 months were excluded because their 5-year outcome was not observed. Patients with missing, invalid, conflicting, or otherwise unsupported required outcome fields were also excluded. The resulting class proportions describe the outcome-eligible analytic samples and should not be interpreted as population survival estimates.

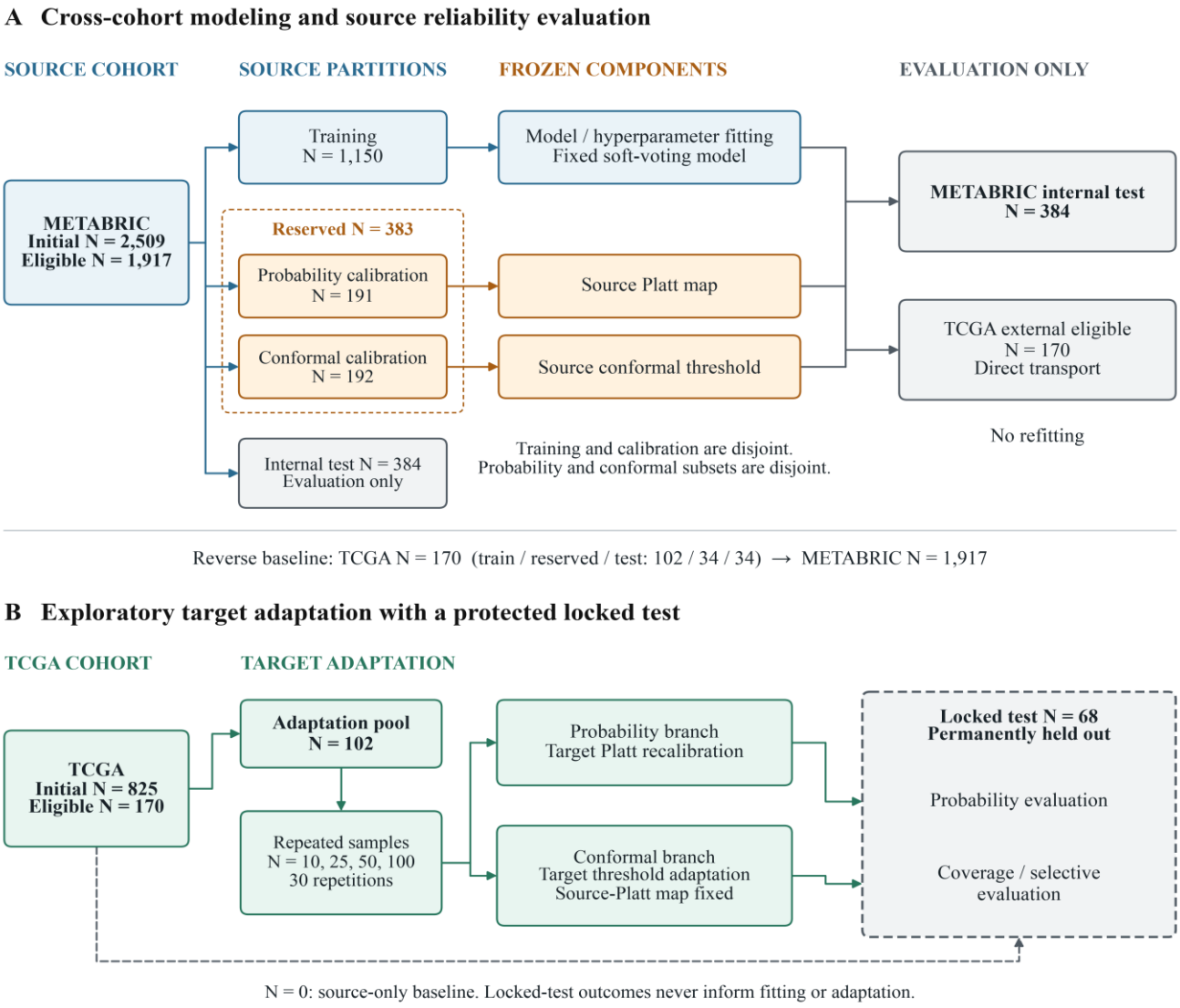


Fig. 1. Study design and protected information boundaries. (A) Bidirectional cross-cohort validation and detailed METABRIC-to-TCGA source calibration and conformal evaluation. (B) Exploratory few-shot target adaptation used a 102-patient TCGA adaptation pool; the 68-patient locked test was reserved for evaluation only.

C. Predictor Harmonization

Candidate shared clinical variables were audited before mapping. Age was parsed in years from AGE_AT_DIAGNOSIS in METABRIC and AGE in TCGA; nonnumeric values or values outside 0–120 years were treated as missing. ER_STATUS, PR_STATUS, and HER2_STATUS were mapped only when the observed value was exactly Positive or Negative. Unknown, equivocal, indeterminate, unavailable, unsupported, conflicting, or absent values were treated as missing.

Tumor stage was not included in primary modeling. METABRIC numeric stages 1–4 could be mapped to I–IV, but TCGA TUMOR_STAGE values represented T categories rather than supported overall stages. Consequently, harmonized stage was missing for all 170 outcome-eligible TCGA patients. The final predictor set was age at diagnosis, estrogen-receptor status, progesterone-receptor status, and HER2 status.

D. Source Partitions and Preprocessing

Random seed 42 was used for source modeling. Within each eligible source cohort, a stratified split allocated approximately 60% to training, 20% to a reserved calibration partition, and 20% to an untouched internal test. METABRIC contributed 1,150 training, 383 reserved calibration, and 384 internal-test patients. TCGA contributed 102, 34, and 34 patients, respectively. Hyperparameter selection used only the training split; the internal test and all external patients were not used for fitting or selection.

Preprocessing was encapsulated in the model pipeline. Age underwent median imputation and standardization. Receptor variables underwent most-frequent imputation followed by one-hot encoding with unknown categories ignored at transformation. Every preprocessing operation was fitted within the applicable training data, including separately within each cross-validation fold, and then applied without refitting to calibration and evaluation patients.

E. Models and Hyperparameter Selection

Five model families were evaluated: logistic regression, radial-basis-function support vector machine with probability output, random forest, XGBoost, and an equal-weight soft-voting ensemble of the four component models. Component hyperparameters were selected by five-fold stratified cross-validation within the source training split, using ROC-AUC as the optimization metric. The small grids varied logistic-regression regularization strength and class weighting; support-vector regularization, kernel scale, and class weighting; random-forest tree count, maximum depth, and minimum leaf size; and XGBoost tree count, depth, and learning rate. The soft-voting model averaged the four fitted component probabilities with weights 1:1:1:1. All stochastic estimators used seed 42.

F. Reproducibility Details

Exact five-fold source-training grids were as follows: logistic regression C∈{0.1, 1.0, 10.0} and class_weight∈{None, balanced}; RBF-SVM C∈{0.5, 1.0, 2.0}, gamma∈{scale, 0.1}, and class_weight∈{None, balanced}; random forest n_estimators∈{100, 200}, max_depth∈{None, 5}, and min_samples_leaf∈{1, 5}; and XGBoost n_estimators∈{100, 200}, max_depth∈{2, 3}, and learning_rate∈{0.05, 0.1}. XGBoost subsample and colsample_bytree were fixed at 0.8. Selection used source-training ROC-AUC only.

Selected METABRIC-source settings were logistic regression C=1.0 with balanced weights; RBF-SVM C=1.0, gamma=0.1, balanced; random forest 100 trees, depth 5, minimum leaf 5; and XGBoost 100 trees, depth 2, learning rate 0.05. Selected TCGA-source settings were logistic regression C=10.0, balanced; RBF-SVM C=2.0, gamma=scale, balanced; random forest 200 trees, depth 5, minimum leaf 5; and XGBoost 100 trees, depth 2, learning rate 0.1. Each soft-voting model used the selected components with weights 1:1:1:1.

The verified environment was Python 3.11.9 with numpy 2.4.6, pandas 3.0.5, scipy 1.17.1, scikit-learn 1.9.0, xgboost 3.2.0, matplotlib 3.11.1, statsmodels 0.14.6, joblib 1.5.3, and requests 2.34.2. Age parsing, receptor mappings, training-only imputation, one-hot encoding, and scaling are specified above and implemented within fitted pipelines. The source Platt map recorded slope 0.907921 and intercept 0.470539; these are transformation coefficients, not external calibration-slope or calibration-intercept performance measures.

Clinical records for brca_metabric and brca_tcga_pub were retrieved on 9 August 2026 (UTC) from https://www.cbioportal.org/api/ using study-specific clinical-attributes and clinical-data endpoints; no separate server API-version string was recorded. The reproducibility archive includes deterministic extraction and harmonization logic, aggregate results, and SHA-256 patient-manifest hashes rather than patient identifiers in the paper. The local repository had no recorded commit and no verified paper-specific public URL or DOI at camera-ready preparation.

G. Internal and External Validation

Each fitted source model was evaluated on its source internal test and directly on all outcome-eligible patients in the other cohort. No target outcome was used for preprocessing, hyperparameter selection, calibration, threshold choice, or model selection. Metrics were ROC-AUC, precision-recall AUC, accuracy, balanced accuracy, sensitivity, specificity, F1 score, Matthews correlation coefficient, and Brier score. The positive class was y_5yr=1. The fixed 0.5 probability threshold was used as a descriptive operating point and was not selected or optimized for a clinical decision context. After inspection of internal descriptive performance, the METABRIC soft-voting model—whose observed internal-test ROC-AUC was highest among the five METABRIC-source models—was carried forward for the detailed calibration, conformal, domain-shift, and few-shot analyses. This was a descriptive, post hoc model-specific choice; these secondary analyses are not used to support a comparative superiority claim.

H. Probability Calibration

Calibration analysis used the reproduced METABRIC soft-voting model and the identical source partitions and component hyperparameters. The 383-patient METABRIC reserved calibration partition was divided, stratified with seed 42, into 191 patients for probability calibration and 192 for conformal calibration. A logistic Platt map was fitted on the logit of the uncalibrated soft-voting probability using only the 191-patient probability-calibration subset. Probabilities were clipped to 10^-6 and 1−10^-6 only for logit transformation. Uncalibrated and Platt-scaled predictions were evaluated on the METABRIC internal test and all 170 eligible TCGA patients using ROC-AUC, Brier score, ten-bin expected calibration error (ECE), and log loss. Reliability plots display observed outcome frequency against mean predicted probability with bin counts (Fig. 3).

I. Split Conformal Prediction and Abstention

The Platt-scaled model was fixed before conformal calibration. For each of the 192 conformal-calibration patients, the nonconformity score was one minus the probability assigned to the observed class. At alpha=0.10, the finite-sample corrected empirical quantile used rank ceil((n+1)(1-alpha)). A candidate class was included when its nonconformity score did not exceed the resulting threshold. The same source-derived threshold was applied to the METABRIC internal test and all eligible TCGA patients.

Outcomes were empirical coverage, mean prediction-set size, singleton rate, defer rate, and accuracy among singleton sets. Selective prediction was operationalized through conformal singleton status: singleton sets yielded a class prediction, whereas empty or two-class sets were deferred. Because target exchangeability was not assumed, TCGA coverage was interpreted as empirical external coverage rather than a guaranteed quantity [6,7].

J. Domain-Shift Analysis

A separate cohort-membership classifier was used to distinguish eligible METABRIC from eligible TCGA patients using only age and ER, PR, and HER2 status. The classifier was a class-balanced logistic-regression pipeline with the same training-fitted preprocessing rules. Five-fold stratified cross-validation with seed 42 produced ROC-AUC and balanced accuracy for cohort discrimination. No survival outcome was used. Descriptive comparisons included age, receptor positivity among nonmissing observations, and predictor missingness.

K. Exploratory Few-Shot Target Adaptation

The 170 eligible TCGA patients were split with stratification and seed 2026 into an adaptation pool of 102 patients and a permanently locked test of 68 patients. The locked test was not used for fitting, calibration, threshold selection, model selection, or choice of target sample size. It contained 22 y_5yr=0 and 46 y_5yr=1 patients.

Target sample sizes were N=0, 10, 25, 50, and 100. N=0 used the source Platt map and source conformal threshold. For each N greater than zero, a stratified sample was drawn from the adaptation pool in 30 repetitions using fixed seeds. Probability adaptation fitted a target Platt map, including slope and intercept, on the sampled patients' fixed METABRIC model scores and evaluated the resulting probabilities on the unchanged locked test. The cancer classifier and its preprocessing were never retrained.

Target conformal adaptation was evaluated as a parallel branch. The source model and source Platt probability function remained fixed, and only the conformal threshold was re-estimated from the N labeled target patients. This target-derived threshold was then evaluated on the same locked test using the fixed source-Platt probabilities. Thus, target labels influenced the probability-recalibration parameters in one branch or the conformal threshold in the other, but never the cancer predictor or locked-test evaluation choices.

L. Protocol Deviations and Exploratory Analyses

The original protocol intent was retained as the confirmatory plan: target sample sizes N = 0, 25, 50, 100, 200 and intercept-only target recalibration. The executed few-shot analysis differed by using N = 0, 10, 25, 50, 100 and fitting both a slope and an intercept through Platt recalibration. The N = 10 condition and fitted slope were not in the initial plan, and N = 200 was not feasible because the realized adaptation pool contained 102 patients. The executed few-shot analysis is therefore designated exploratory and post hoc. Its results are hypothesis-generating and are not treated as confirmatory evidence for the original specification.

The information boundary was nevertheless preserved. The 68-patient locked test remained disjoint from the 102-patient adaptation pool and untouched by model fitting, probability recalibration, conformal-threshold estimation, threshold or sample-size selection, and stopping decisions. The fixed locked test was used only for evaluation after those choices had been set, and the cancer classifier was never retrained.

Probability recalibration and conformal adaptation were separate exploratory branches. Target Platt recalibration evaluated probability calibration by fitting slope and intercept on the sampled target patients. Conformal target adaptation did not use the target-Platt probabilities: it updated only the nonconformity threshold using the sampled target labels while keeping the cancer model and source-Platt probabilities fixed.

M. Statistical Analysis

Point estimates and 95% confidence intervals for ROC-AUC, precision-recall AUC, Brier score, sensitivity, and specificity were computed using 2,000 patient-level nonparametric bootstrap replicates and percentile limits [3,4]. The same resampled patient indices were used across models evaluated on a common cohort, permitting paired comparisons. Internal-minus-external ROC-AUC differences and external-minus-internal Brier differences were calculated from bootstrap distributions. Pairwise model comparisons were performed for ROC-AUC and Brier score within each evaluation setting; two-sided bootstrap P values were adjusted within each family using Holm's method [26]. Conformal coverage received Wilson 95% confidence intervals.

For few-shot adaptation, each N greater than zero was summarized across 30 target-sampling repetitions by the mean, standard deviation, and 2.5th and 97.5th percentiles. These percentiles quantify variation due to the sampled adaptation patients while holding the 68-patient locked test fixed; they are not patient-level confidence intervals for a new target population. Analyses used Python 3.11 and the fixed software environment recorded in the project.

IV. Results

A. Cohorts and Outcome Eligibility

METABRIC contained 2,509 patients, of whom 1,917 had an observable binary 5-year outcome; 427 (22.27%) were classified as death by 60 months and 1,490 (77.73%) as known survival beyond 60 months. Of 592 excluded patients, 64 were alive or censored by 60 months and 528 lacked valid overall-survival months. TCGA contained 825 patients, of whom 170 were outcome eligible; 55 (32.35%) were classified as death by 60 months and 115 (67.65%) as known survival beyond 60 months. Of 655 exclusions, 648 were alive or censored by 60 months, and seven lacked valid overall-survival months.

Among eligible patients, mean age was 61.19 years in METABRIC and 57.22 years in TCGA. ER-positive proportions among nonmissing observations were 76.21% and 70.55%, PR-positive proportions were 52.97% and 60.74%, and HER2-positive proportions were 12.32% and 13.10%, respectively. TCGA predictor missingness was 4.12% for ER, 4.12% for PR, and 14.71% for HER2, compared with 0%, 0.05%, and 0.05% in METABRIC. Cohort characteristics are presented in Table I.

TABLE I. Cohort Characteristics for the Outcome-Eligible Analytic Samples. Receptor Percentages use Nonmissing Denominators

Cohort	Initial N	Eligible N	Death by 60 months	Survival beyond 60 months	Age, mean (SD)	ER positive	PR positive	HER2 positive	Missing ER / PR / HER2
METABRIC	2,509	1,917	427 (22.27%)	1,490 (77.73%)	61.19 (12.94)	76.21%	52.97%	12.32%	0.00% / 0.05% / 0.05%
TCGA	825	170	55 (32.35%)	115 (67.65%)	57.22 (14.42)	70.55%	60.74%	13.10%	4.12% / 4.12% / 14.71%

B. Bidirectional Internal and External Performance

For METABRIC-source models, the soft-voting ensemble had the highest observed internal ROC-AUC: 0.732 (95% CI, 0.672–0.786) on 384 internal-test patients. Its ROC-AUC was 0.643 (95% CI, 0.557–0.728) on all 170 eligible TCGA patients (Table II; Fig. 2). The observed internal-to-external numerical decrease was approximately 0.088 (95% CI, -0.015 to 0.196). Because this interval included zero, the decrease was not statistically significant. Brier score increased from 0.160 (95% CI, 0.142–0.177) internally to 0.214 (95% CI, 0.182–0.247) externally, an external-minus-internal difference of 0.054 (95% CI, 0.020–0.090). Precision-recall AUC was 0.903 internally and 0.768 externally.

For TCGA-source models, the RBF support vector machine had the highest observed internal ROC-AUC: 0.674 (95% CI, 0.460–0.861) on 34 patients. Applied to all 1,917 eligible METABRIC patients, its ROC-AUC was 0.586 (95% CI, 0.554–0.615). The observed internal-to-external numerical decrease was also approximately 0.088 (95% CI, -0.127 to 0.282), and this interval likewise included zero. Its Brier score was 0.206 (95% CI, 0.155–0.263) internally and 0.183 (95% CI, 0.177–0.190) externally, an external-minus-internal difference of -0.023 (95% CI, -0.080 to 0.028). The TCGA-source internal estimate is substantially less precise because the source internal test contains only 34 patients; the two transport directions therefore should not be interpreted as providing symmetric statistical evidence.

Threshold-based performance was uneven. The METABRIC-source soft-voting model had sensitivity 0.956 and specificity 0.174 internally, and sensitivity 0.930 and specificity 0.109 in TCGA. Several RBF support-vector evaluations assigned every patient to the positive class at the fixed descriptive 0.5 operating point, yielding sensitivity 1.000 and specificity 0.000; the zero specificity reflects this fixed operating point and is not a recommended clinical threshold. Complete metrics and bootstrap intervals for all five models are provided in Table II. Holm-adjusted pairwise model-comparison results are retained in the verified statistical output, but the approximately 0.088 cross-cohort numerical decreases above are interpreted from their confidence intervals and are not labeled significant.

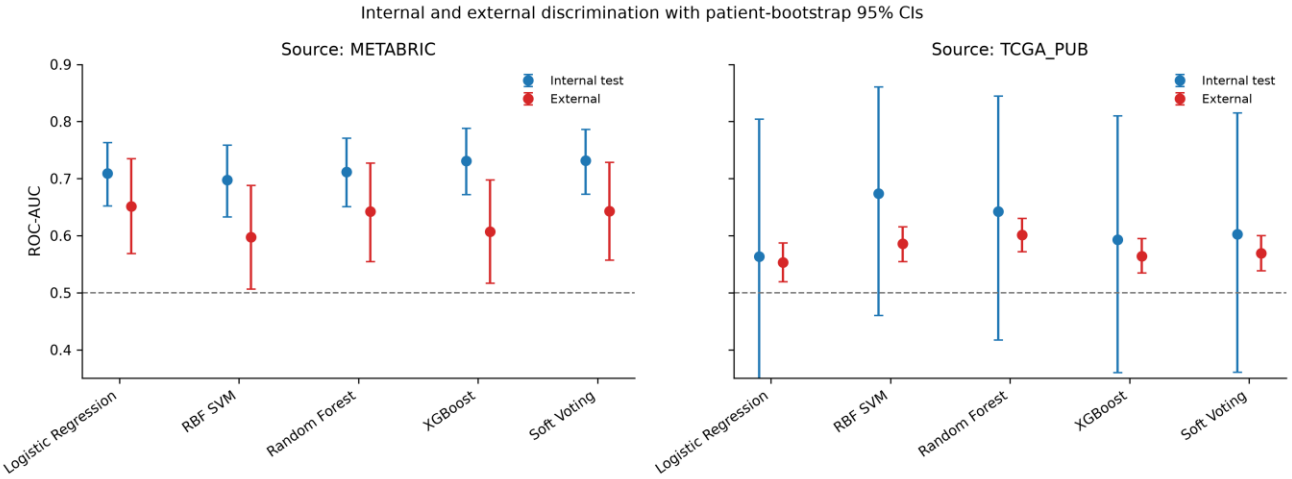


Fig. 2. Internal and external ROC-AUC for both transport directions with 95 percent bootstrap confidence intervals.

TABLE II. BIDIRECTIONAL INTERNAL AND EXTERNAL PERFORMANCE. THE POSITIVE CLASS WAS KNOWN SURVIVAL BEYOND 60 MONTHS; SENS. DENOTES SENSITIVITY AND SPEC. SPECIFICITY AT THE FIXED 0.5 THRESHOLD

Source	Evaluation	Model	N	ROC-AUC [95% CI]	PR-AUC [95% CI]	Brier [95% CI]	Sens.	Spec.
METABRIC	METABRIC Internal test	Logistic Regression	384	0.709 [0.652–0.763]	0.901 [0.868–0.930]	0.228 [0.215–0.241]	0.695	0.581
METABRIC	METABRIC Internal test	RBF SVM	384	0.698 [0.633–0.758]	0.878 [0.836–0.918]	0.161 [0.139–0.183]	1.000	0.000
METABRIC	METABRIC Internal test	Random Forest	384	0.712 [0.651–0.770]	0.888 [0.847–0.924]	0.156 [0.133–0.178]	0.963	0.151
METABRIC	METABRIC Internal test	XGBoost	384	0.731 [0.672–0.788]	0.895 [0.855–0.930]	0.152 [0.130–0.173]	0.993	0.105
METABRIC	METABRIC Internal test	Soft Voting	384	0.732 [0.672–0.786]	0.903 [0.867–0.933]	0.160 [0.142–0.177]	0.956	0.174
METABRIC	TCGA External eligible	Logistic Regression	170	0.651 [0.568–0.735]	0.819 [0.744–0.883]	0.235 [0.213–0.256]	0.713	0.455
METABRIC	TCGA External eligible	RBF SVM	170	0.597 [0.506–0.688]	0.751 [0.657–0.842]	0.223 [0.185–0.263]	1.000	0.000
METABRIC	TCGA External eligible	Random Forest	170	0.643 [0.554–0.727]	0.788 [0.699–0.864]	0.221 [0.183–0.265]	0.957	0.091
METABRIC	TCGA External eligible	XGBoost	170	0.607 [0.517–0.698]	0.746 [0.652–0.837]	0.226 [0.185–0.270]	0.957	0.127
METABRIC	TCGA External eligible	Soft Voting	170	0.643 [0.557–0.728]	0.768 [0.672–0.862]	0.214 [0.182–0.247]	0.930	0.109

TCGA	TCGA Internal test	Logistic Regression	34	0.563 [0.300–0.804]	0.698 [0.517–0.904]	0.231 [0.161–0.311]	0.739	0.455
TCGA	TCGA Internal test	RBF SVM	34	0.674 [0.460–0.861]	0.811 [0.639–0.954]	0.206 [0.155–0.263]	1.000	0.000
TCGA	TCGA Internal test	Random Forest	34	0.642 [0.417–0.844]	0.793 [0.626–0.938]	0.207 [0.140–0.283]	0.957	0.091
TCGA	TCGA Internal test	XGBoost	34	0.593 [0.360–0.810]	0.746 [0.568–0.922]	0.210 [0.124–0.306]	0.913	0.273
TCGA	TCGA Internal test	Soft Voting	34	0.603 [0.360–0.815]	0.768 [0.595–0.920]	0.205 [0.141–0.277]	0.957	0.364
TCGA	METABRIC External eligible	Logistic Regression	1,917	0.553 [0.519–0.587]	0.786 [0.761–0.813]	0.303 [0.292–0.314]	0.517	0.581
TCGA	METABRIC External eligible	RBF SVM	1,917	0.586 [0.554–0.615]	0.812 [0.787–0.838]	0.183 [0.177–0.190]	1.000	0.000
TCGA	METABRIC External eligible	Random Forest	1,917	0.601 [0.571–0.630]	0.838 [0.816–0.861]	0.191 [0.184–0.198]	0.799	0.218
TCGA	METABRIC External eligible	XGBoost	1,917	0.564 [0.534–0.595]	0.817 [0.794–0.840]	0.228 [0.218–0.239]	0.715	0.349
TCGA	METABRIC External eligible	Soft Voting	1,917	0.569 [0.538–0.600]	0.813 [0.790–0.837]	0.213 [0.205–0.220]	0.755	0.314

C. Calibration and Conformal Transport

On the METABRIC internal test, source Platt scaling preserved ROC-AUC at 0.732 while changing Brier score from 0.160 to 0.157, ECE from 0.066 to 0.044, and log loss from 0.494 to 0.485. In TCGA, ROC-AUC remained 0.643, but source-only Platt scaling worsened the Brier score from 0.214 to 0.222, ECE from 0.078 to 0.104, and log loss from 0.616 to 0.641. Fig. 3 shows the corresponding reliability plots, and Table III reports the calibration and conformal metrics. Thus, the source calibration map showed a small favorable internal change but did not transport as an improvement in external calibration.

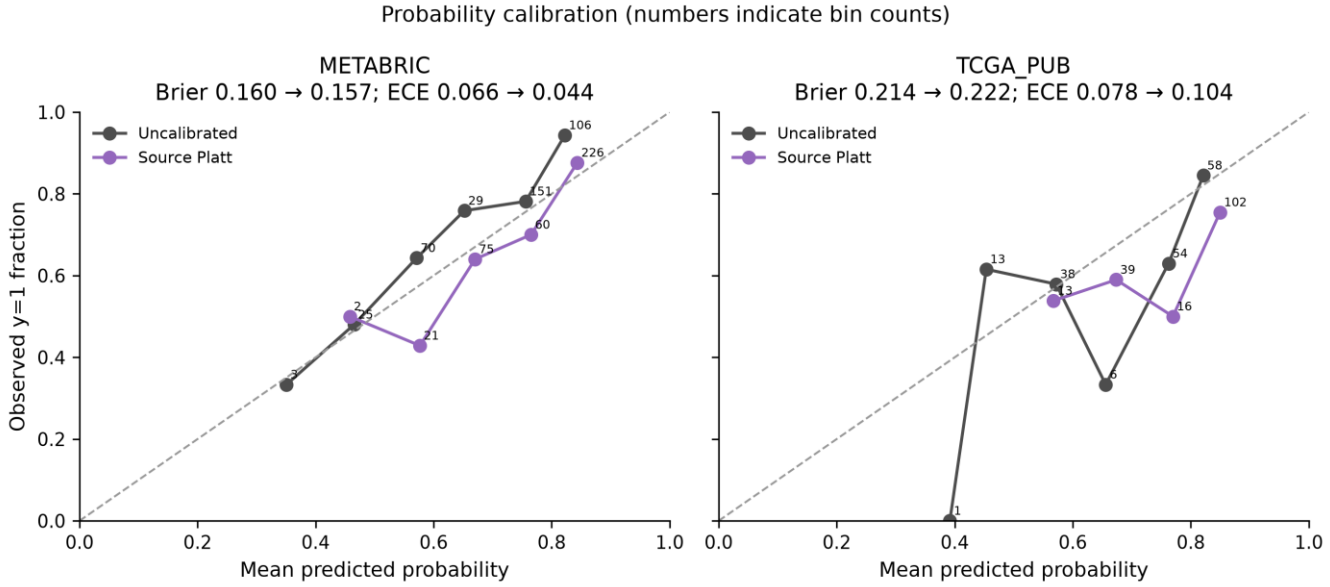


Fig. 3. Calibration behavior of the fixed METABRIC soft-voting model before and after source Platt scaling.

At nominal 90% coverage, empirical conformal coverage was 0.891 (Wilson 95% CI, 0.855–0.918) on the METABRIC internal test and 0.818 (Wilson 95% CI, 0.753–0.868) in TCGA (Fig. 4). The external interval did not include 0.900. Mean prediction-set size was 1.302 internally and 1.329 externally. Singleton and defer rates were 0.698 and 0.302 internally, compared with 0.671 and 0.329 externally. Accuracy among singleton predictions was 0.843 internally and 0.728 externally. These results show external undercoverage of the source-derived conformal sets.

TABLE III. SOURCE CALIBRATION AND SPLIT-CONFORMAL TRANSPORT AT ALPHA 0.10. ECE USED TEN EQUAL-WIDTH BINS. TCGA COVERAGE IS EMPIRICAL EXTERNAL COVERAGE

Evaluation	N	Uncalibrated AUC	Uncalibrated Brier / ECE / log loss	Platt Brier / ECE / log loss	Coverage [Wilson 95% CI]	Mean set size	Singleton	Defer	Singleton accuracy
METABRIC internal	384	0.732	0.160 / 0.066 / 0.494	0.157 / 0.044 / 0.485	0.891 [0.855–0.918]	1.302	0.698	0.302	0.843
TCGA external	170	0.643	0.214 / 0.078 / 0.616	0.222 / 0.104 / 0.641	0.818 [0.753–0.868]	1.329	0.671	0.329	0.728

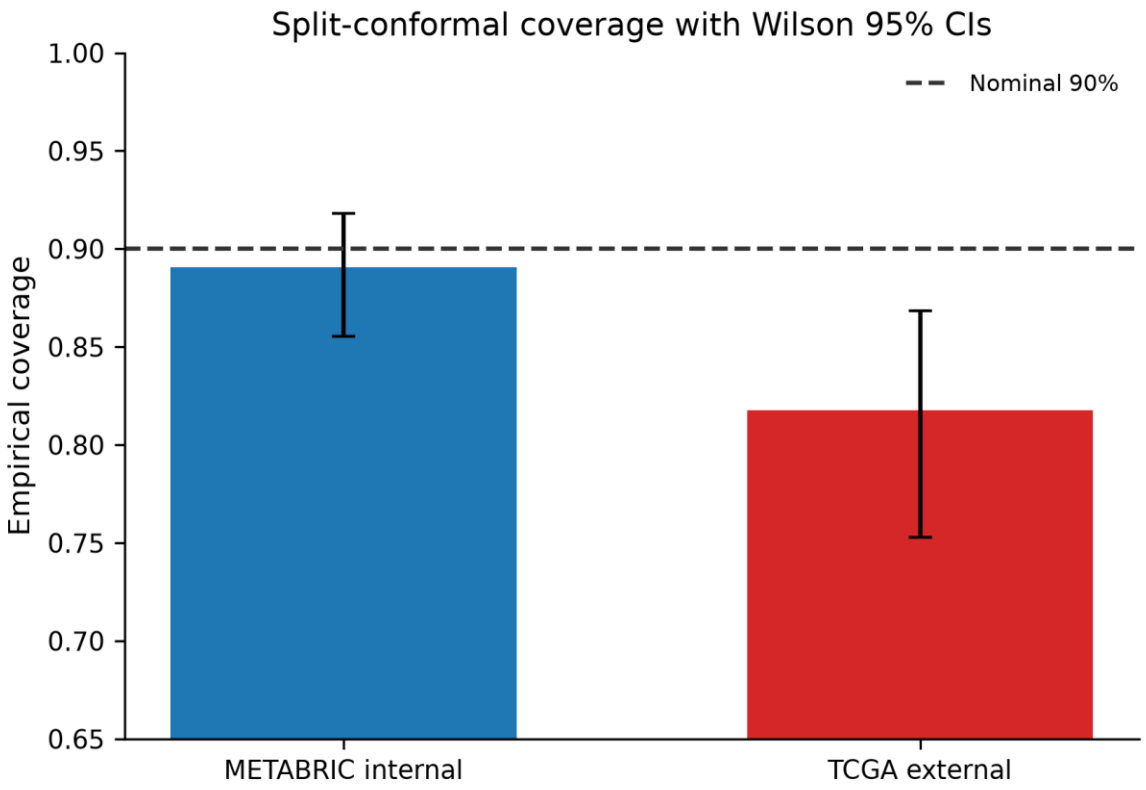


Fig. 4. Internal and external empirical split-conformal coverage relative to the nominal 90 percent level.

D. Domain Shift

Eligible TCGA patients were younger on average than METABRIC patients by 3.97 years, with a standardized mean difference of -0.290. Receptor prevalence and missingness also differed, most notably for HER2 missingness. The cohort-membership classifier achieved mean five-fold ROC-AUC 0.604 (SD, 0.049) and balanced accuracy 0.574 (SD, 0.047). These label-free results indicate measurable, although incomplete, separability using the four harmonized predictors and their missingness patterns.

E. Exploratory Few-Shot Target Adaptation

The TCGA adaptation pool contained 102 patients (33 y_5yr=0, 69 y_5yr=1), and the locked test contained 68 patients (22 and 46, respectively). Table IV and Fig. 5 summarize repeated target adaptation. With N=0, the locked-test Brier score was 0.226, and ECE was 0.128. At N=10, mean Brier score was 0.254 (SD, 0.059) and mean ECE was 0.162 (SD, 0.109), showing instability at the smallest target sample. Mean Brier score and ECE were 0.219 and 0.066 at N=25, 0.214 and 0.047 at N=50, and 0.213 and 0.036 at N=100. Relative to N=0, larger target samples therefore improved both locked-test Brier score and ECE without retraining the cancer classifier. Singleton accuracy at N=10 was not summarized because three of the 30 repetitions produced no singleton predictions, making per-repetition singleton accuracy undefined and causing the frozen all-repetition aggregation to return NA.

Target conformal adaptation did not restore mean nominal coverage. Mean coverage was 0.809 at N=0, 0.882 at N=10, 0.886 at N=25, 0.866 at N=50, and 0.868 at N=100. The highest mean was 0.886 at N=25, below 0.900. Mean defer rates were 0.294, 0.543, 0.554, 0.469, and 0.456 across the same N values. Fig. 5 displays Brier score, ECE, and conformal coverage with 2.5th–97.5th percentiles and the nominal 90% reference line.

TABLE IV. EXPLORATORY FEW-SHOT TARGET ADAPTATION ON THE FIXED 68-PATIENT TCGA LOCKED TEST. VARIATION IS ACROSS TARGET SAMPLES, NOT PATIENT-BOOTSTRAP CONFIDENCE INTERVALS

Target N	Repetitions	Brier mean (SD)	ECE mean (SD)	Log loss mean (SD)	Coverage mean (SD)	Mean set size	Defer mean (SD)	Singleton accuracy mean (SD)
0	1	0.226 (0.000)	0.128 (0.000)	0.653 (0.000)	0.809 (0.000)	1.294	0.294 (0.000)	0.729 (0.000)
10	30	0.254 (0.059)	0.162 (0.109)	1.426 (2.632)	0.882 (0.065)	1.543	0.543 (0.240)	NA
25	30	0.219 (0.009)	0.066 (0.047)	0.633 (0.025)	0.886 (0.044)	1.554	0.554 (0.172)	0.740 (0.029)
50	30	0.214 (0.001)	0.047 (0.020)	0.618 (0.003)	0.866 (0.013)	1.469	0.469 (0.048)	0.747 (0.012)
100	30	0.213 (0.000)	0.036 (0.009)	0.615 (0.000)	0.868 (0.000)	1.456	0.456 (0.000)	0.757 (0.000)

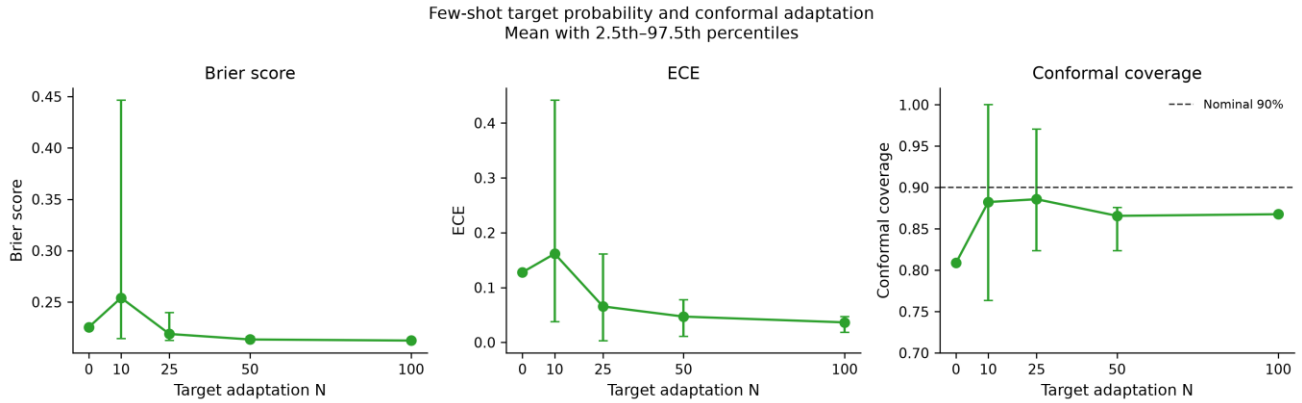


Fig. 5. Exploratory few-shot target recalibration and conformal threshold adaptation on the fixed TCGA locked test.

V. Discussion

The main finding was an observed numerical decrease in discrimination in both transport directions. At the reported precision, the METABRIC soft-voting model and the highest-observed-AUC TCGA internal model each showed an internal-to-external ROC-AUC difference of approximately 0.088. This numerical similarity is descriptive: both 95% confidence intervals included zero, so neither decrease was statistically established. The TCGA-source internal estimate is substantially less precise because its internal test contained only 34 patients; the two directions therefore provide complementary evidence about directionality but not symmetric statistical evidence. The soft-voting model was carried forward for detailed uncertainty analysis only after inspection of internal descriptive performance. Accordingly, those model-specific analyses are secondary and do not support a claim of prospectively selected superiority.

Probability calibration exposed a different aspect of transportability. Source Platt scaling improved the observed METABRIC internal Brier score, ECE, and log loss, but the same fixed map produced worse observed values for all three metrics in TCGA. ROC-AUC was unchanged because the fitted transformation preserved score ordering. Discrimination and calibration therefore answered different questions. These metric changes show that the source map did not transport as an observed improvement in TCGA; the ten-bin ECE alone is not treated as proof of general calibration failure. Source-only recalibration should be evaluated externally as part of the transported prediction system. It cannot be assumed to be universally corrective.

The conformal analysis showed a parallel limitation of source-derived uncertainty after transport. Empirical coverage was 89.1% on the METABRIC internal test, close to the nominal 90%, but 81.8% in TCGA. The TCGA Wilson interval ended at 86.8%, and external evaluation also showed a higher defer rate and lower singleton accuracy. Split-conformal validity depends on an exchangeability relationship between calibration and evaluation observations; that relationship was not established across these cohorts. The result is therefore external empirical undercoverage, not a failure of conformal theory or a contradiction of the source guarantee.

The label-free domain analysis provided context without identifying a cause. A cohort-classifier ROC-AUC of 0.604 showed that METABRIC and TCGA were partly distinguishable using the four predictors and their missingness patterns. TCGA patients were younger on average, receptor distributions differed, and HER2 missingness was higher. These findings are compatible with covariate shift, but they do not attribute the performance changes to any single variable or exclude unmeasured differences in selection, follow-up, or care. Because the shift analysis was not used to select features or models, it informs interpretation rather than comparative model ranking.

The exploratory, post hoc few-shot analysis separated probability recalibration from conformal threshold adaptation. Target Platt recalibration improved mean locked-test Brier score and ECE at N=25, 50, and 100 relative to N=0, with the lowest means at N=100. N=10 was worse and highly variable. Estimating both a calibration slope and intercept from very limited labels was unstable. These probability results suggest that labeled target data can correct aspects of the score-to-risk relationship, but they remain conditional on one 68-patient locked test and do not establish a generally effective adaptation sample size.

The conformal branch answered a narrower and separate question. It kept the cancer model and source-Platt probabilities fixed and updated only the conformity threshold using target adaptation patients. This simple threshold update did not restore mean 90% coverage at any evaluated N; the highest mean coverage was 88.6% at N=25, and coverage was 86.8% at N=100. The lack of monotonic recovery indicates that a target-derived threshold alone was insufficient in this small experiment. It does not compare alternative conformal methods and cannot support a superiority claim about any uncertainty procedure.

These results concern the transportability of the complete prediction workflow. Internal discrimination did not determine external discrimination, and internal calibration improvement did not imply external calibration improvement. Near-nominal internal conformal coverage also did not imply target coverage. Cross-cohort evaluation should keep target data protected and label post hoc adaptation explicitly. Ranking, probability accuracy, and set coverage should be reported separately. The

present findings do not establish clinical usefulness or deployment readiness.

A. Future Work

Future external-validation studies should use larger cohorts and clinically prespecified operating thresholds, with threshold uncertainty and decision-curve/net-benefit analysis. Calibration assessment should add calibration intercept and slope, flexible calibration curves with uncertainty bands, and a prevalence-only Brier reference. Richer harmonization should examine overall stage where semantically defensible, principled treatment of cohort-specific HER2 missingness, and molecular or other prognostic features.

Direct comparison with established clinical prognostic tools such as PREDICT and the Nottingham Prognostic Index should be undertaken when their required inputs can be reconstructed consistently across cohorts. A natural extension is a sequential target-adaptation pipeline in which target probability recalibration is followed by conformal recalibration using disjoint target data. Censoring-aware approaches, including Cox regression, random survival forests, and survival boosting, should be evaluated to use partially observed follow-up more efficiently while preserving a clinically interpretable horizon; such methods do not by themselves eliminate selection concerns. Temporal, geographic, prospective, and workflow-based validation also remain necessary.

VI. Limitations

First, the outcome-eligible TCGA cohort contained only 170 patients; its source internal test contained 34, and the few-shot locked test contained 68. These sample sizes yielded wide confidence intervals and limited the resolution of performance and adaptation estimates. In particular, the TCGA-source direction has lower precision and should not receive evidentiary weight symmetric to the METABRIC-source direction. Reusing one locked test across 30 target-sampling repetitions enabled controlled comparisons, but the reported percentiles are conditional on that fixed test and quantify adaptation-sample variation rather than test-population uncertainty.

Second, only 170 of 825 TCGA patients and 1,917 of 2,509 METABRIC patients met the binary 5-year outcome rule. In TCGA, 648 patients were excluded because they were alive or censored by 60 months, and seven had invalid or missing overall-survival months. The resulting complete-outcome analysis may represent a selected subset; no censoring-aware survival model was fitted.

Third, the shared predictor set contained only age and ER, PR, and HER2 status. This deliberate restriction enabled transparent cross-cohort semantic alignment but excluded important prognostic and molecular information. Tumor stage was not used because TCGA TUMOR_STAGE represented T categories, and no supported overall-stage mapping was available; cohort-specific receptor, particularly HER2, missingness also remained. Alternative stage or HER2 harmonization strategies were not evaluated.

Fourth, the detailed calibration, conformal, domain-shift, and few-shot analyses were performed only for METABRIC-to-TCGA transport using the soft-voting model; equivalent reverse-direction uncertainty analyses were not performed. The reported ten-bin ECE is a descriptive, bin-dependent calibration summary and is particularly sensitive to finite-sample variation in the 170-patient external TCGA cohort. Calibration intercept and slope as external-validation measures, flexible calibration curves with uncertainty bands, and comparison with a prevalence-only Brier reference were not part of the accepted analysis. Brier-score and log-loss results remain complementary but do not remove these limitations.

Fifth, the fixed 0.5 threshold was only a descriptive operating point. Threshold-dependent sensitivity and specificity, including zero specificity in several RBF-SVM evaluations, must not be interpreted as recommended clinical operating points. Although frozen patient-bootstrap outputs contain confidence intervals for sensitivity and specificity at 0.5, no clinical threshold optimization or uncertainty analysis around a clinically chosen threshold was performed, and no decision-curve or net-benefit analysis was conducted.

Finally, target probability recalibration estimated both slope and intercept and was unstable at very small N. The accepted exploratory design evaluated probability recalibration and conformal-threshold adaptation as separate branches to isolate their respective effects while maintaining the locked evaluation boundary; a combined probability-plus-conformal target-adaptation pipeline was not evaluated. Direct comparison with PREDICT or the Nottingham Prognostic Index was also outside scope because the required inputs could not be reconstructed consistently from the deliberately restricted harmonized predictor set. This retrospective two-cohort study included no prospective, temporal, geographic, workflow-embedded, subgroup-fairness, or clinical-impact validation. Clinical deployment readiness cannot be inferred.

VII. Conclusion

Using age and receptor status, machine-learning models showed observed numerical decreases in ROC-AUC in both METABRIC-to-TCGA and TCGA-to-METABRIC transport. Both decreases were approximately 0.088, but their 95% confidence intervals included zero. Source-only Platt calibration improved the observed internal probability metrics but produced worse observed external Brier score and ECE. Source split-conformal prediction showed external empirical undercoverage in TCGA, reaching 81.8% rather than nominal 90% coverage. Exploratory few-shot target recalibration improved locked-test Brier score and ECE at larger N, but target conformal-threshold adaptation did not restore mean 90% coverage. These findings support separate evaluation of discrimination, calibration, and uncertainty under cohort shift and do not establish clinical usefulness or deployment readiness.

Credit Authorship Contribution Statement

NADIR Younes: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review and editing, Project administration, Supervision.

AMHAIMAR Lahcen: Methodology, Visualization, Software.

Mohamed Azouazi: Supervision, Validation, Writing – review and editing.

KHALIDI Abderrahim: Supervision, Validation, Writing – review and editing.

RACHDI Mohamed: Methodology, Formal analysis, Writing – review and editing.

BAKHOUYI Abdellah: Methodology, Investigation, Data curation.

DATA AVAILABILITY

This study used patient-level and sample-level clinical data for cBioPortal study identifiers brca_metabric and brca_tcga_pub, retrieved through the official REST API on 9 August 2026 (UTC). No mutation, expression, copy-number, imaging, or complete study-archive data were used. Source clinical data remain available through cBioPortal subject to the originating studies' data-use terms; no raw protected, private, or directly identifying data are redistributed with the reproducibility archive.

CODE AVAILABILITY

The executable analysis workflow comprises numbered Python scripts, fixed environment requirements, machine-readable aggregate result tables, split-manifest hashes, and figure-generation code. A publication-ready release candidate has been prepared for author deposit, but no paper-specific public repository URL, release tag, commit, or archival DOI was verified at camera-ready preparation; consequently, no public identifier is claimed here.

ETHICS STATEMENT

This study was a secondary analysis of publicly available, de-identified clinical data obtained through cBioPortal. No new participants were recruited or contacted, and no directly identifying information was accessed. Ethical approval and informed consent for the original data collection were governed by the respective METABRIC and TCGA study protocols.

FUNDING

This research received no external funding.

CONFLICT OF INTERESTS

The authors declare no conflict of interest.

DECLARATION ON GENERATIVE AI

During the preparation of this work, the authors used ChatGPT (OpenAI) to assist with language refinement and editorial preparation. All AI-assisted checks, numerical results, interpretations, and manuscript content were critically reviewed and validated by the authors. The authors take full responsibility for the content of the publication.

REFERENCES

[1] Bray F, Laversanne M, Sung H, Ferlay J, Siegel RL, Soerjomataram I, Jemal A. Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin. 2024;74(3):229–263. https://doi.org/10.3322/caac.21834

[2] Li J, Zhou Z, Dong J, et al. Predicting breast cancer 5-year survival using machine learning: A systematic review. PLoS One. 2021;16(4):e0250370. https://doi.org/10.1371/journal.pone.0250370

[3] Collins GS, Dhiman P, Ma J, et al. Evaluation of clinical prediction models (part 1): from development to external validation. BMJ. 2024;384:e074819. https://doi.org/10.1136/bmj-2023-074819

[4] Riley RD, Archer L, Snell KIE, et al. Evaluation of clinical prediction models (part 2): how to undertake an external validation study. BMJ. 2024;384:e074820. https://doi.org/10.1136/bmj-2023-074820

[5] Van Calster B, McLernon DJ, van Smeden M, Wynants L, Steyerberg EW. Calibration: the Achilles heel of predictive analytics. BMC Med. 2019;17:230. https://doi.org/10.1186/s12916-019-1466-7

[6] Angelopoulos AN, Bates S. Conformal prediction: a gentle introduction. Found Trends Mach Learn. 2023;16(4):494–591. https://doi.org/10.1561/2200000101

[7] Tibshirani RJ, Barber RF, Candès EJ, Ramdas A. Conformal prediction under covariate shift. In: Advances in Neural Information Processing Systems 32. 2019:2526–2536. https://proceedings.neurips.cc/paper_files/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html

[8] Curtis C, Shah SP, Chin SF, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature. 2012;486:346–352. https://doi.org/10.1038/nature10983

[9] Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. Nature. 2012;490:61–70. https://doi.org/10.1038/nature11412

[10] Hueting TA, van Maaren MC, Hendriks MP, Koffijberg H, Siesling S. External validation of 87 clinical prediction models supporting clinical decisions for breast cancer patients. Breast. 2023;69:382–391. https://doi.org/10.1016/j.breast.2023.04.003

[11] Clift AK, Dodwell D, Lord S, Petrou S, Brady M, Collins GS, Hippisley-Cox J. Development and internal-external validation of statistical and machine learning models for breast cancer prognostication: cohort study. BMJ. 2023;381:e073800. https://doi.org/10.1136/bmj-2022-073800

[12] Elmejgari C, El Khalfaoui Y, Nadir Y, Alibouch B, Qbadou M. A hybrid approach for breast cancer classification using CLAHE-enhanced mammograms and EfficientNetV2S. In: Mejdoub Y, Elamri A, Kardouchi M, editors. Connected Objects, Artificial Intelligence, Telecommunications and Electronics Engineering. Lecture Notes in Networks and Systems, vol. 1584. Cham: Springer; 2026. p. 536–542. https://doi.org/10.1007/978-3-032-01536-5_81

[13] Elmejgari C, Nadir Y, Qbadou M. Deep learning-based techniques for breast cancer classification: a systematic review. In: 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET). IEEE; 2024. p. 1–7. https://doi.org/10.1109/IRASET60544.2024.10548784

[14] Collins GS, Moons KGM, Dhiman P, et al. TRIPOD+AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. BMJ. 2024;385:e078378. https://doi.org/10.1136/bmj-2023-078378

[15] Lee S, Yin C, Zhang P. Stable clinical risk prediction against distribution shift in electronic health records. Patterns. 2023;4(9):100828. https://doi.org/10.1016/j.patter.2023.100828

[16] Rabanser S, Günnemann S, Lipton ZC. Failing loudly: an empirical study of methods for detecting dataset shift. In: Advances in Neural Information Processing Systems 32. 2019:1394–1406. https://proceedings.neurips.cc/paper_files/paper/2019/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html

[17] Lin HT, Lin CJ, Weng RC. A note on Platt's probabilistic outputs for support vector machines. Mach Learn. 2007;68(3):267–276. https://doi.org/10.1007/s10994-007-5018-6

[18] Steyerberg EW, Borsboom GJJM, van Houwelingen HC, Eijkemans MJC, Habbema JDF. Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. Stat Med. 2004;23(16):2567–2586. https://doi.org/10.1002/sim.1844

[19] Binuya MAE, Engelhardt EG, Schats W, Schmidt MK, Steyerberg EW. Methodological guidance for the evaluation and updating of clinical prediction models: a systematic review. BMC Med Res Methodol. 2022;22:316. https://doi.org/10.1186/s12874-022-01801-8

[20] Shashikumar SP, Amrollahi F, Nemati S. Unsupervised detection and correction of model calibration shift at test-time. Annu Int Conf IEEE Eng Med Biol Soc. 2023;2023:1–4. https://doi.org/10.1109/EMBC40787.2023.10341086

[21] Tanner KT, Keogh RH, Coupland CAC, Hippisley-Cox J, Diaz-Ordaz K. Dynamic updating of clinical survival prediction models in a changing environment. Diagn Progn Res. 2023;7:24. https://doi.org/10.1186/s41512-023-00163-z

[22] Barber RF, Candès EJ, Ramdas A, Tibshirani RJ. Conformal prediction beyond exchangeability. Ann Stat. 2023;51(2):816–845. https://doi.org/10.1214/23-AOS2276

[23] Yang Y, Kuchibhotla AK, Tchetgen Tchetgen E. Doubly robust calibration of prediction sets under covariate shift. J R Stat Soc Series B Stat Methodol. 2024;86(4):943–965. https://doi.org/10.1093/jrsssb/qkae009

[24] Geifman Y, El-Yaniv R. SelectiveNet: A deep neural network with an integrated reject option. In: Proceedings of the 36th International Conference on Machine Learning. PMLR. 2019;97:2151–2159. https://proceedings.mlr.press/v97/geifman19a.html

[25] Hendrickx K, Perini L, Van der Plas D, Meert W, Davis J. Machine learning with a reject option: a survey. Mach Learn. 2024;113:3073–3110. https://doi.org/10.1007/s10994-024-06534-x

[26] Holm S. A simple sequentially rejective multiple test procedure. Scand J Stat. 1979;6(2):65–70. https://www.jstor.org/stable/4615733

Unmasking the Unknown: Feature-Orchestrated Image Synthesis for Deep Learning-Based Threat Detection

Sruthi C K, Harini Sriraman*
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India

***Abstract*—IoT cybersecurity is crucial as a vast number of linked, susceptible devices serve as possible entry points into bigger networks. Recently, deep learning and machine learning have been used to detect both known and unknown threats. These models suffer from false positives, outdated datasets, biased evaluations against similar attacks, and an inability to generalize against unknown attacks. This study proposes Feature-Orchestrated Image Synthesis (FOS), a technique that selects key features via an explainable AI method, SHAP, then arranges them by tiling the scaled feature value across the row, and then repeats this process with each feature value to guide the image synthesis. SHAP offers a single framework that can be applied to both local and global interpretations. Benchmark datasets BoT-IoT and CICIoMT2024 were used to evaluate the models, InceptionNet and Vision Transformer (ViT). The models were trained only on Denial-of-Service attack and normal traffic images. Our proposed SHAP-ordered approach effectively prioritizes high-influence features, ensuring dependable performance and consistent detection accuracy even in complex, unbalanced network settings. Using the ViT and InceptionNet architectures, it significantly improves detection capabilities over the full features baseline across both CICIoMT2024 and BoT-IoT datasets, yielding recall improvements of 50.2% for ARP attacks and 10.8% for reconnaissance on ViT, along with a 19.0% boost in OS Fingerprint recall on InceptionNet.**

***Keywords*—Unknown attack; pretrained CNN; vision transformer; feature selection; correlation coefficient; SHAP**

I. Introduction

Attacks against the Internet of Things (IoT) take advantage of vulnerabilities in devices linked to the internet in order to obtain illegal access, disrupt operations, and steal data. Attackers have two options for breaking into a network: known attacks and unknown attacks. Security systems are able to identify known threats [1] such as malware and Denial of Service (DoS) attacks. Unknown attacks are novel or modified attack techniques that security systems have never seen before. They are also known as zero-day attacks or exploits. Attacks have long been detected using deep learning and machine learning algorithms. Deep learning techniques are effective at identifying new or zero-day attacks where traffic trends shift from the training distribution, even though both machine learning and deep learning models can successfully identify known attacks by identifying pre-observed patterns. Moreover, deep learning systems are naturally better at scaling to manage massive amounts of high-dimensional network traffic. For unknown attack detection, the models are trained on a particular type of attack and tested on different types of attacks. These attacks may be related to the ones the model has been trained upon, or there may also be a few attacks that are dissimilar to the trained ones. Any type deviating from the training input can be considered unknown. In this way, most of the previous works have given good performance against attacks that are either variants of or attacks that are similar to the training set. In the case of attacks that are unrelated, comparatively, the models have not been able to give decent performance. Both image and non-image forms of inputs to identify the pattern have been given to the models. The performance drawback with respect to unrelated unknown attacks points towards a lack of generalization capability of the models. Feature selection is included under feature engineering. Feature selection techniques include filter methods, wrapper methods, and embedded methods [2]. Additionally, IDS processes large amounts of data quickly, which inevitably increases the processing cost to determine whether the data is malicious or benign. Effective and fast feature selection techniques are therefore required, especially in order to identify zero-day threats. In order to optimize IDS datasets, feature selection is essential. The computational load on IoT devices can be greatly decreased by identifying and keeping only the most important features that affect detection accuracy. By removing unnecessary or redundant data, lowering dimensionality, and accelerating the learning process, it improves model performance. Among the various advantages of feature selection is an improvement in generalization capability by overcoming overfitting. Previous works related to unknown attack detection have highlighted results where different attacks from the same family of attacks [3] technically qualify for the term unknown attack. But these models tend to fail when given an unknown attack from a different family. There is also the case of false positives for the models, wherein normal traffic gets misclassified as an attack. Machine learning models [4] exhibit good performance, but they lack the ability to scale to large volumes of network data if implemented in real time. In order to detect zero-day attacks, this work employs Feature Orchestrated Image Synthesis (FOS), followed by deep neural networks, InceptionNet, and Vision Transformer (ViT). Shapley Additive Explanation (SHAP) is used in the FOS, an Explainable AI (XAI) approach to select important features from the dataset, arrange them by tiling the scaled feature value across the row, and then repeat this procedure with each feature value to direct the picture synthesis process. While LIME (Local Interpretable Model-Agnostic Explanations), another XAI technique, only provides

*Corresponding author.

local interpretation, the SHAP technique has the advantage of providing both global and local interpretation. When compared to scenarios where all features were employed or where the correlation coefficient technique was used for feature selection, the suggested approach produced better detection performance. The contributions of this study are as follows:

- Feature-orchestrated Image Synthesis by using the XAI method, SHAP for selecting key features, arranging them by tiling the scaled feature values across the rows, and then repeating this step for all the feature values, thereby guiding the image synthesis process.
- The features are structured spatially according to their functional characteristics (e.g., statistical metrics, packet headers, and control flags).

The study has the following sections: Section II contains related work, Section III contains methodology followed by results and discussion, and the last section gives the conclusion and future work.

II. Related Works

A. Known Attacks

Using Autoencoders, Saranya, K. Et Al. [5] Carry out Cross-layer Attack Detection of Assaults Linked to Several Network Layers, Such as DDoS and Man-in-the-Middle (MiTM). in This Study, an Autoencoder Has Been Employed for Both Anomaly Detection and Feature Extraction. a Comparison of ML and DL Models Was Presented in the Study [6]. Although DL Models Outperform ML Models, Their Precision, Recall, and F1 Score Only Range from 65 to 70%, Which Is Unsatisfactory. The use of Hybrid Models, Such as Support Vector Machine (SVM)for Multi-class Classification and Deep Belief Network (DBN) for Feature Extraction, in Conjunction with Adversarial Training, Was Suggested in the Study [7].

B. Unknown Attacks

Al-Zewairi et al. [8] proposed the use of a multi-stage, multi-layer zero trust architecture that combines supervised and unsupervised learning stages. The study made use of DBSCAN (Density-Based Spatial Clustering of Applications with Noise), an unsupervised machine learning model. One of the drawbacks of this model is that a single set of parameters can be applied if all clusters are of equal/uniform density throughout the dataset, and it is not scalable. Using Pearson's Correlation Coefficient (PCC) for feature selection, Hakami et al. [9] discovered that RF and Long Short-Term Memory (LSTM) outperformed other models when evaluated on the WSN-DS, UNSW-NB15, and CIC-IDS2017 datasets. Using Enhanced Grey Wolf Optimization (EGWO) for feature selection, Alqahtany et al. [10] proposed random forest (RF) for intrusion detection on the NF-ToN-IoT dataset, achieving an accuracy of 99.93% by choosing 23 features from the dataset. In a different study, Akuthota et al. [11] assessed their attention-based transformer using the NSL-KDD and UNSW-NB15 datasets, achieving 99% accuracy in binary and multi-class classification. The RTIoT2022 dataset was used by Benmalek et al. [12] for a number of models, including CNN, LSTM, K-Nearest Neighbors (KNN), Naïve Bayes (NB), and Categorical Boosting (CatBoost). Out of all the models, CatBoost did the best. In this work, optimization and feature selection were also used. XGBoost (XGB) was proposed by Bikila et al. [13] as a classifier and an autoencoder for feature extraction. On the CICIoT2023 and CICIDS2017 datasets, it attained an accuracy of 99%. A hybrid ML-DL framework that includes CNN and decision trees was used by M. R. Minhas et al. [14] to detect distributed denial of service (DDoS) attacks. For the CICDDoS2019 and CICIDS2017 datasets, they attained an accuracy of 96%. A hybrid ML-DL framework that includes CNN and decision trees was used. For the CICDDoS2019 and CICIDS2017 datasets, the accuracy was 96%. Combining CNN with LSTM models, where CNN is used for feature extraction and LSTM is a classifier, was suggested by Smmarwar et al. [15]. Here, the model identified many kinds of malware families. A hybrid model of soft ordering-based CNN (SOCNN) with outlier identification methods was applied to the BoT-IoT, CICIDS-2017, and CIC-IDS-2018 datasets in study [16]. Its maximum average F1 score was 98.94%. L1 and L2 regularization with CNN was used in another study by Hairab et al. [17] to boost generalizability to identify unidentified threats. They used just nine features from the BoT-IoT dataset. Bot-IoT and N-BaIoT datasets have been subjected to federated learning (FL) [18], which has the benefit of protecting privacy. It was employed to detect zero-day botnet attacks. Both machine learning and deep learning models have been used to identify known and unknown threats on IoT networks. Despite this, both types have demonstrated good performance for a variety of benchmark datasets when employed separately or in an ensemble mode. The inability of machine learning algorithms to handle changing data patterns and scale enormous amounts of data is a disadvantage. Some of the investigations have made use of out-of-date simulation-based datasets that may not reflect network activity in real time. Additionally, it has been noted that certain works have detected unknown attacks that have similarities to the attack that is included in the training set. In our work, feature-orchestrated image synthesis is done by using the XAI method, SHAP, to select key features, followed by arranging the feature values by tiling the scaled feature values across the rows and repeating this process for every selected feature, thus guiding the image synthesis process. The model's performance has been compared to scenarios in which feature selection was not used, utilizing the correlation coefficient and earlier SOTA studies.

Our work introduces a more stringent cross-category assessment methodology, whereas earlier studies have often tested models on unknown attacks that show substantial behavioral similarities or are in the same family as the training distribution. In particular, our models are trained solely on volumetric Denial-of-Service (DoS) behavior and tested on structurally unrelated protocols and intrusion types, such as ARP spoofing, reconnaissance, and OS fingerprinting, where the underlying traffic profiles and feature value distributions are fundamentally different, instead of testing on intra-family variants.

III. Methodology

The following section explains the steps taken to implement the proposed work.

A. System Description

1) Dataset description: In this proposed research, two datasets have been utilized, CICIoMT2024[19] and BoT-IoT [20]. Both datasets have been created by the Canadian Institute of Cybersecurity. A specialized subset called the CICIoMT2024 dataset concentrates on the particular setting of the Internet of Medical Things (IoMT).

- Researchers at UNSW Canberra Cyber developed the BoT-IoT dataset, a large-scale, labeled network benchmark dataset. It contains attacks such as DoS, DDoS, reconnaissance, and information theft. For each network flow, the dataset monitors 46 different statistical and behavioral characteristics.
- A practical benchmark dataset for Internet of Medical Things (IoMT) devices is the CICIoMT2024 dataset. This dataset was created utilizing an IoMT testbed of 40 IoMT devices and included attack categories: mainly DDoS, DoS, reconnaissance, and spoofing. It includes 47 features. All the features are numerical data.

Table I gives the number of samples across attack and normal sets of individual datasets.

The steps followed in the proposed methodology are as follows:

Step 1: Dataset preprocessing: Following feature selection, removal of NaN or any undefined/unrepresented values. Also performed label encoding to convert categorical values to numerical ones, if any are present within the dataset.

Step 2: Feature-Orchestrated Image Synthesis (FOS): Step 2.1: Feature Importance: To select the most relevant, non-redundant, and informative subset of features from the dataset, feature selection techniques have been applied. Feature selection also helps in increasing the unseen attack resilience of the model against unknown attacks. Correlation coefficient [21] and SHAP (Shapley Additive Explanation) [22] techniques were applied to perform feature selection.

- Correlation Coefficient: Features from the dataset are chosen using correlation coefficient values related to the type of assault. We used a two-stage feature selection pipeline to improve model interpretability and decrease multicollinearity: first, redundant features were eliminated by finding pairs with an absolute correlation coefficient greater than 0.90; second, we kept only those features that showed a significant linear relationship with the target variable, as determined by an absolute correlation threshold of 0.1. Based on the correlation coefficient, 12 relevant features for the CICIoMT2024 dataset and 7 relevant features for the BoT-IoT dataset were chosen.
- SHAP, a component of eXplainable AI (XAI), aids in understanding the model's predictions. The model, it offers both local and global forecasts. Another XAI model is LIME; however, it focuses on how the model is interpreted locally. Only the training data—consisting of normal traffic and Denial-of-Service (DoS) attacks—is fed into a tree-based model (Random Forest Classifier) to calculate the SHAP values. The feature significance values derived from this training subset are then used to choose the most important characteristics. Using these SHAP feature importance values, the 12 most relevant features from CICIoMT2024 and BoT-IoT are selected for further processing.

Functionality-based feature ordering: Network traffic metrics can be methodically arranged into coherent functional categories by combining feature rankings obtained from both linear statistical dependencies (like correlation coefficients) and game-theoretic attribution techniques (like SHAP). This domain-aware taxonomy organizes variables into basic operational dimensions instead of depending on isolated, unrestricted feature lists: Packet timing and pacing are captured by Temporal & Duration Metrics; traffic intensity and cumulative byte movement are measured by Volume & Sequence Metrics; low-level connection states and header properties are defined by Structural & Protocol Identity; and complex multi-feature interactions, TCP control flags, and connection distributions are modelled by Statistical & Aggregated Behavior. By connecting the underlying network mechanics with the raw algorithmic output, this structured alignment improves model interpretability and reduces multicollinearity within operational groups. Table II summarises the features selected based on the Correlation Coefficient and SHAP model.

Step 2.2: Train-validation and test split: The training and validation set contains DoS attacks and normal traffic, while the testing set includes the remaining attacks available in the dataset. The train-validation split is a random stratified split that is applied to the pool of normal traffic and DoS attack samples using scikit-learn's train_test_split with the stratify option at an 80:20 ratio. While training using the BoT-IoT dataset, since there is an imbalance between normal and attack traffic, the penalty weight concept has been applied. A penalty weight that is inversely proportional to class frequencies is computed. The neural network is forced to prioritize learning its distinctive characteristics equally because minority classes are given a higher weight. This means that if the model incorrectly classifies a rare attack sample during training, the loss penalty is significantly increased.

Step 2.3: Image Synthesis: In order to create an image, the selected feature values are scaled to a range between 0 and 255. Each scaled feature value is taken and assigned to every pixel across the entire row of the image. This means that an ith feature value is repeated over the entire ith row. This way, a pattern is formed with the selected features. The images are stored with files named "Attack" or "Normal." This is followed by training, validation, and testing sets.

A deterministic image synthesis pipeline is used to convert the tabular network traffic records into spatial visual representations appropriate for Deep Learning architectures. To ensure robustness against unexpected attack outliers, feature vectors are first scaled and clipped into the range [0.0, 1.0] using global minimum and maximum values that are only generated from the training set. The features are sorted in a functionality-based sequence according to their declining SHAP feature relevance score for a given sample with N

selected features. The initial $(N \times N)$ 2D grid matrix, where the i-th feature value uniformly spans the whole i-th row, is created by replicating the scaled features' $(N \times 1)$ column vector across N columns using pixel repetition.

This grid is mapped through the perceptually uniform viridis colormap to create a 3-channel RGB image representation and the necessary 96x96x3 RGB tensor. The image is accurately scaled to 96x96 pixels using nearest-neighbor interpolation (Image.NEAREST) because the feature dimension N does not divide 96 evenly. In order to maintain the structural integrity of the spatial patterns, this resampling technique is essential since it maintains distinct boundaries and stops color leakage across feature rows. The generated photos are then classified as "Attack" or "Normal" and placed in the appropriate directories for testing, validation, and training.

Step 3: Classification model: Pretrained CNN and Vision Transformer models have been used for training and testing the models against attacks that are different from the training ones. The number of epochs for training the models has been decided through the early stopping technique. In the case of the BoT-IoT dataset, there exists an imbalance with respect to attack and normal traffic in the training set; hence, penalties based on class weights are applied to prevent the model from being biased towards the minority class. Vision transformer: Vision Transformers (ViT) [23] have recently emerged as a competitive alternative to the state-of-the-art (SOTA) Convolutional Neural Networks (CNNs) in a range of image recognition and computer vision tasks. The self-attention mechanism must be incorporated into the transformer architecture in order to detect contextual information and long-range linkages in the incoming data. Because a ViT model relies on a self-attention mechanism, it can concentrate on different portions of the input data. Transformer blocks, which include a multilayer perceptron (MLP) head, patch embedding, positional embedding, and a multi-head self-attention layer, are the fundamental elements of ViT. In this work, a patch size of 16 is employed. This particular patch size was selected because it evenly divides the 96×96 picture dimensions, creating a structured 6×6 token grid that is consistent with normal ViT configurations and efficiently captures both global spatial relationships and local feature groupings. This last layer has been modified to produce a binary classification result. InceptionNet: Also called GoogLeNet [24]. With many convolution filter sizes (1 x 1, 3 x 3, and 3 x 3, 5 x 5), the "inception module" was introduced. The computational costs are also reduced by this. InceptionNet is available in multiple versions. Because InceptionNet is computationally efficient, reduces overfitting, and improves accuracy, it outperforms other pretrained models. These pretrained models can be modified to meet the demands of a particular task.

Fig. 1 depicts the flowchart of the entire proposed method. Image_CorrCoeff represents correlation coefficient-based image synthesis, and Image_SHAP follows SHAP feature importance for image synthesis of attack and normal traffic. Table III gives details about various characteristics of the models. Table IV lists the hardware and software specifications used for this work.

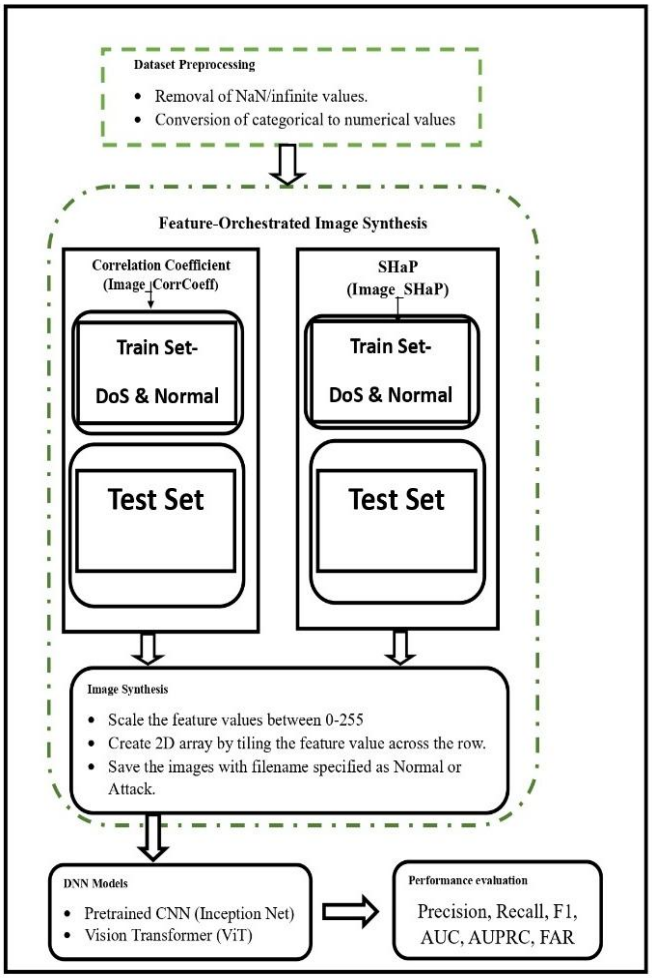


Fig. 1. Flowchart for proposed work.

TABLE I. NUMBER OF SAMPLES OF ATTACK/NORMAL TRAFFIC IN DATASETS

Dataset	Category	DoS	ARP Spoofing	Reconnaissance	OS FingerPrint	Service Scan
CICIoMT2024	Attack	10000	1744	>10000	-	-
	Normal	10000	5000	>2000	-	-
BoT-IoT	Attack	10000	-	-	>10000	10000
	Normal	200	-	-	100	100

TABLE II. FEATURE LIST BASED ON DIFFERENT FEATURE SELECTION TECHNIQUES

Dataset	Feature selection	Features
BoT-IoT	Correlation Coefficient	pkSeqID, proto_number, state, dur, TnP_PerProto, N_IN_Conn_P_SrcIP, N_IN_Conn_P_DstIP
	SHAP	ltime, stime, pkSeqID, pkts, spkts, state, proto_number, TnP_Per_Dport, Pkts_P_State_P_Protocol_P_DestIP, TnP_PDstIP,TnP_PerProto,TnBPDstIP
CICIoMT2024	Correlation Coefficient	Variance, Max, Std,AVG, Min, TotSum, Rate,Header_Length,HTTPS,ProtocolType, ARP,psh_flag_number,rst_count,syn_count
	SHAP	Number, Max, AVG, Min, Variance, IAT, Weight, Header_Length, Tot size, ack_flag_number,psh_flag_number, Magnitude

TABLE III. INCEPTIONNET AND VIT MODEL SPECIFICATIONS

Characteristics	InceptionNet	ViT
Image Size	96 x 96 x 3	
Batch Size	**32**	
No.of parameters	23.8 million	8.07 million
Activation Function	Sigmoid in the output layer	GELU in MLP block; Sigmoid in output layer
Optimizer	Adam	AdamW
EarlyStopping	**Patience=10, Monitor=validation loss**	
Reduce LR onPlateau	Patience=5, factor=0.5, min lr=1e-6	Patience=5, factor=0.5, min lr=1e-7

TABLE IV. SYSTEM CONFIGURATION AND SOFTWARE LIBRARIES UTILIZED

Item	Specification	Library Name	Usage
Workstation	Lenovo Ideapad	Keras	Building and training neural network
OS	Windows 11	Numpy	Data preprocessing
System Version	25H2	Pandas	Data manipulation and analysis
OSbuild	26200.7462	Tensor Flow	Numerical computations
Processor	AMD Ryzen 3		
RAM	GB		

IV. RESULTS AND DISCUSSION

The outcomes of the models for classification utilizing images made with various feature subsets according to the number of features are explained in this section. The outcomes have been contrasted with cutting-edge research. Balanced accuracy, precision, recall, False Alarm Rate (FAR), and Receiver Operating Characteristic (ROC)-AUC value are the performance metrics [29] used for assessment.

- Balanced accuracy: When working with unbalanced datasets—that is, where one class significantly outnumbers the others—a performance metric known as balanced accuracy is utilized to evaluate classification models. The arithmetic mean of Sensitivity (True Positive Rate) and Specificity (True Negative Rate) is known as balanced accuracy.
- Recall (True Positive Rate, Sensitivity): Recall gauges the system's ability to recognize each actual attack. The formula is True Positives/ (True Positives + False Negatives).
- FAR (False Alarm Rate): Also referred to as the False Positive Rate (FPR), it measures how frequently the system misidentifies normal, benign traffic as a hostile attack. FAR stands for False Positives/Actual Negatives.
- ROC-AUC value: The ROC AUC score is a statistic that may be used to summarize a classifier's performance across multiple thresholds. The score is calculated by measuring the area under the ROC curve. The ROC AUC score ranges from 0 to 1. Perfect execution is given a score of 1, while random guessing is given a score of 0.5. If a model's score is slightly higher than 0.5, it is considered to have at least "some" (although limited) predictive power. In general, this is not enough for any real-world applications.

TABLE V. EVALUATION BASED ON PREVIOUS WORKS CICIOMT2024 DATASET

Method	Year	Performance Evaluation	Remarks
[25]	2026	Multi-stage evaluation (AE, Binary, Categorical, Multiclass accuracy/success rates. XGBoost achieves a 90.87% success rate on the held-out attack.	1. Framework for multi-stage learning to identify both known and unknown IoMT attacks. 2. Combines hierarchical attack attribution with swift anomaly detection. 3. Assessed using open-set novelty handling in IoMT security scenarios.
[26]	2025	Accuracy: 97.31%	1. Computational overhead is high. 2. 1D CNN is used. 3. Not A Zero-day work. 4. Examined multiple datasets.
[27]	2026	AUROC-OOD of 0.975 with an FPR@95%TPR of 7.8%	1. Open-set recognition 2. Zero-day threat

			3. Evaluation on unseen testbeds. 4. Focuses on time-series protocol compliance over 1D-to-2D spatial image transformations.
[28]	2025	Validation accuracy of 99.23% for CICIoMT2024	1. Focuses strictly on sequential time-series modelling and temporal dependencies. 2. Bidirectional Gated Recurrent Unit (BiGRU), Long Short-Term Memory (LSTM), and Multi-Head Attention (MHA) for sequential time-series feature modelling.
Proposed Work		Balanced Accuracy: 1. For ViT (ARP spoofing): 84.00%, and reconnaissance attack is 90%.	1. Zero-day model. 2. Achieves a +25.86% (InceptionNet) and +47.37% (ViT) balanced accuracy improvement over unordered baselines via SHAP ordering. 3. Balanced accuracy improved via SHAP ordering.

TABLE VI. PERFORMANCE METRICS COMPARISON FOR CICIOMT2024 ARP ATTACK CASES (IN-INCEPTIONNET)

Method	Balanced accuracy (%)		Recall for attack (%)		FAR		AUC		AUPRC		Precision of attack (%)		F1 Score of attack (%)	
	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT
All features	58	57	56	19	0.39	0.06	0.6	0.57	0.39	0.33	33	54	42	29
Correlation Coefficient Unordered	56	56	30	19	0.19	0.07	0.47	0.39	0.28	0.28	36	47	33	27
SHAP Unordered	68	68	98	37	0.63	0	0.74	0.68	0.58	0.53	35	98	52	53
Correlation Coefficient Ordered	74	61	59	28	0.11	0.05	0.75	0.45	0.64	0.39	65	65	62	39
SHAP Ordered	73	84	54	77	0.07	0.09	0.71	0.87	0.67	0.8	72	75	62	76

TABLE VII. PERFORMANCE METRICS COMPARISON FOR CICIOMT2024 RECONNAISSANCE ATTACK CASES (IN-INCEPTIONNET)

Method	Balanced accuracy (%)		Recall for attack (%)		FAR		AUC		AUPRC		Precision of attack (%)		F1 Score of attack (%)	
	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT
All features	88	82	89	65	0.14	0.03	0.87	0.71	0.99	1	99	100	94	79
Correlation Coefficient Unordered	93	90	91	80	0.06	0.01	0.96	0.86	1	0.99	99	100	95	89
SHAP Unordered	78	68	63	37	0.07	0	0.89	0.68	0.99	0.95	99	100	77	54
Correlation Coefficient Ordered	94	71	91	43	0.03	0.04	0.97	0.71	1	0.99	100	100	95	60
SHAP Ordered	78	90	57	87	0.01	0.06	0.81	0.94	0.98	0.99	100	99	73	93

TABLE VIII. PERFORMANCE EVALUATION WITH PREVIOUS WORK FOR BOT-IOT

Method	Year	Performance Evaluation	Remarks
[30]	2025	99.60% accuracy,23 features used	Not a zero-day model; ML models RF and SVM were used.
[17]	2022	9 features used. Acc:99% for DDoS, 90% for reconnaissance attack.	CNN with regularizations applied. Non-image input was used. Work related to a zero-day attack.
Proposed		Absolute Metrics (SHAP Ordered): OS Fingerprint: Balanced Accuracy: 99.00% (IN) / 100.00% (ViT); Recall: 99.00% (IN) / 100.00% (ViT). Service Scan: Balanced Accuracy: 86.00% (IN) / 94.00% (ViT); Recall: 76.00% (IN) / 100.00% (ViT).	Image-based input. SHAP-ordered feature arrangement provides better results compared to the unordered case.

TABLE IX. PERFORMANCE METRICS COMPARISON FOR BOT-IOT_OS FINGERPRINT ATTACK CASES (IN-INCEPTIONNET)

Method	Balanced accuracy (%)		Recall for attack (%)		FAR		AUC		AUPRC		Precision of attack (%)		F1 Score of attack (%)	
	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT
All features	65	100	98	100	0.69	0	0.57	1	1	1	100	100	99	100
Correlation Coefficient Unordered	93	99	86	98	0	0	0.93	0.99	1	1	100	100	92	99
SHAP Unordered	100	100	100	100	0	0	1	1	1	1	100	100	100	100
Correlation Coefficient Ordered	98	100	96	100	0	0	0.97	1	1	1	100	100	98	100
SHAP Ordered	99	100	99	100	0.02	0	0.98	1	1	1	100	100	100	100

TABLE X. PERFORMANCE METRICS COMPARISON FOR BOT-IOT SERVICE SCAN ATTACK CASES (IN-INCEPTIONNET)

Method	Balanced accuracy (%)		Recall for attack (%)		FAR		AUC		AUPRC		Precision of attack (%)		F1 Score of attack (%)	
	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT	IN	ViT
All features	89	100	77	100	0	0	0.96	1	1	1	100	100	87	100
Correlation Coefficient Unordered	51	82	2	64	0	0	0.6	0.82	0.99	1	100	100	4	78
SHAP Unordered	98	97	100	97	0.04	0.03	0.98	0.99	1	1	100	100	100	99
Correlation Coefficient Ordered	82	98	76	98	0.13	0.02	0.79	0.99	1	1	100	100	86	99
SHAP Ordered	86	94	76	100	0.04	0.12	0.96	0.99	1	1	100	100	87	100

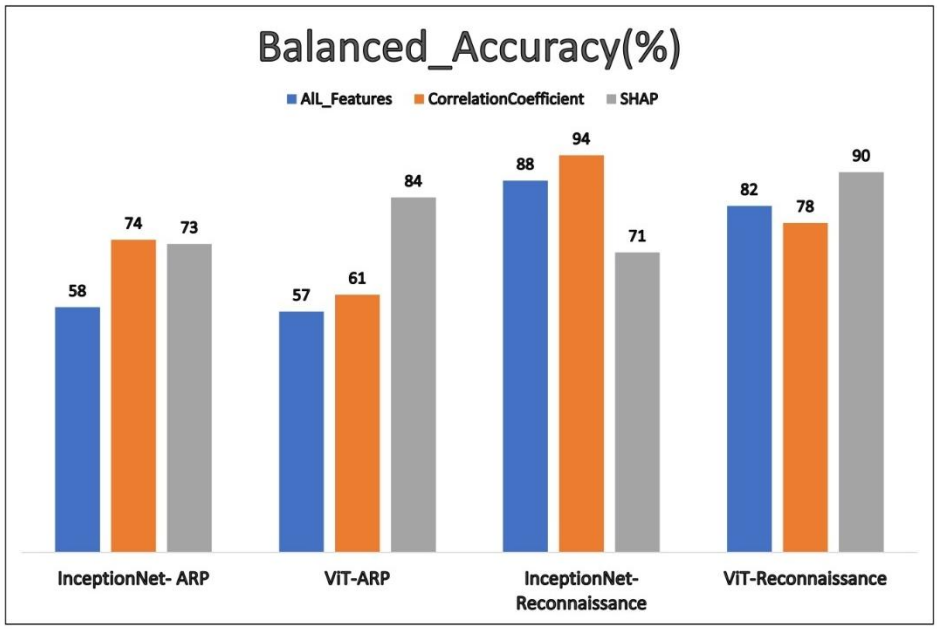


Fig. 2. Balanced accuracy for CICIoMT2024 dataset.

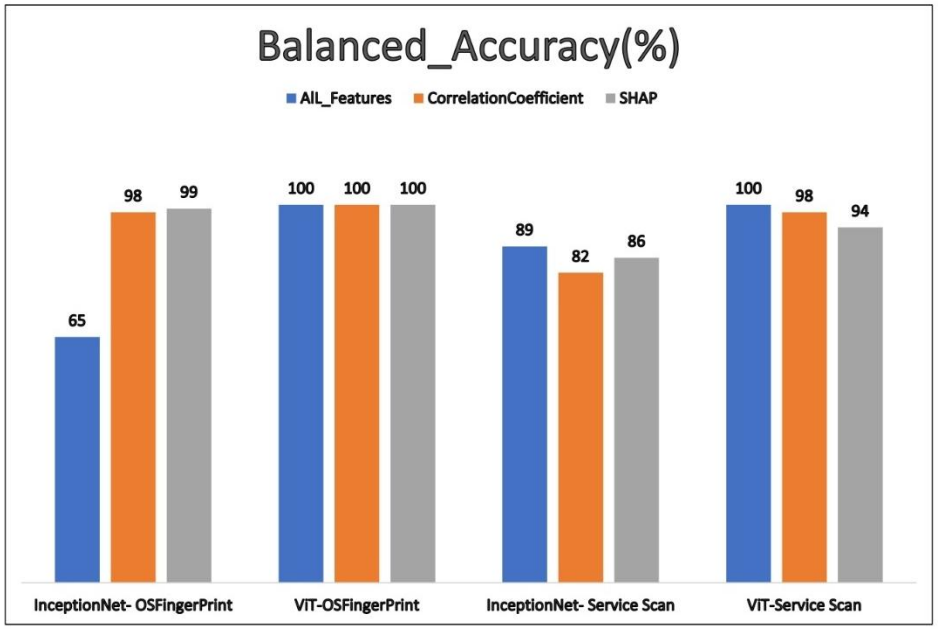


Fig. 3. Balanced accuracy for BoT-IoT dataset.

TABLE XI. MULTIPLE SEED EVALUATION OF CICIOMT2024 DATASET TESTED ON DIFFERENT ATTACKS

Attacks	Balanced accuracy (%)	Recall for attack (%)	FAR	AUC	AUPRC	Precision of attack (%)	F1 Score of attack (%)
InceptionNet							
All Features: Arp Spoof	59.00 ± 3.74	52.60 ±7.70	0.35 ±0.05	0.60 ±0.06	0.42 ±0.07	34.60 ±3.65	41.60 ±4.39
All Features: Reconnaissance	81.80 ±4.15	80.40 ±10.45	0.16 ±0.05	0.84 ±0.04	0.984 ±0.005	98.40 ±0.55	87.00 ±6.36
SHAP ordered: ARP spoof	73.60 ±0.89	52.40 ±2.19	0.05 ±0.02	0.73 ±0.02	0.68 ±0.01	78.60 ±4.51	62.60 ±0.89
SHAP ordered: Reconnaissance	79.20 ±1.64	64.00 ±9.35	0.054 ±0.059	0.818 ±0.011	0.98 ±0.00	99.40 ±0.89	77.80 ±6.30
ViT							
All Features: ARP Spoof	67.20 ±9.36	43.80 ±24.08	0.10 ±0.05	0.63 ±0.16	0.45 ±0.15	60.60 ±4.39	48.60 ±15.34
All Features: Reconnaissance	84.20 ±5.50	76.00 ±9.49	0.08 ±0.06	0.84 ±0.09	0.99 ±0.01	99.20 ±0.84	86.00 ±5.83
SHAP ordered: ARP Spoof	78.60 ±6.35	65.80 ±12.38	0.086 ±0.021	0.792 ±0.087	0.702 ±0.103	72.00 ±6.71	68.40 ±8.93
SHAP ordered: Reconnaissance	87.60 ±5.03	84.20 ±7.09	0.084 ±0.050	0.90 ±0.055	0.988 ±0.008	99.00 ±0.71	91.00 ±4.18

TABLE XII. MULTIPLE SEED EVALUATION OF BOT-IOT DATASET TESTED ON DIFFERENT ATTACKS

Attacks	Balanced accuracy (%)	Recall for attack (%)	FAR	AUC	AUPRC	Precision of attack (%)	F1 Score of attack (%)
InceptionNet							
All Features: OS Fingerprint	64.00 ±2.45	83.00 ±32.33	0.488 ±0.318	0.68 ±0.185	1.00 ±0.00	100.00 ±0.00	86.80 ±26.06
All Features: Service Scan	85.60 ±6.84	78.60 ±11.50	0.072 ±0.070	0.90 ±0.13	1.00 ±0.00	100.00 ±0.00	87.40 ±7.16
SHAP ordered: OS Fingerprint	98.60 ±0.55	98.80 ±0.45	0.02 ±0.00	0.986 ±0.009	1.00 ±0.00	100.00 ±0.00	99.40 ±0.59
SHAP ordered: Service Scan	85.40 ±0.55	76.00 ±0.00	0.06 ±0.014	0.954 ±0.005	1.00 ±0.00	100.00 ±0.00	87.00 ±0.00
ViT							
All Features: OS Fingerprint	100.00 ±0.00	99.60 ±0.55	0.00 ±0.00	1.00 ±0.00	1.00 ±0.00	100.00 ±0.00	100.00 ±0.00
All Features: Service Scan	95.00 ±7.55	92.40 ±15.04	0.024 ±0.035	0.944 ±0.106	1.00 ±0.00	100.00 ±0.00	95.60 ±9.13
SHAP ordered: OS Fingerprint	100.00 ±0.00	100.00 ±0.00	0.00 ±0.00	1.00 ±0.00	1.00 ±0.00	100.00 ±0.00	100.00 ±0.00
SHAP ordered: Service Scan	88.00 ±8.54	86.00 ±19.16	0.096 ±0.039	0.97 ±0.019	1.00 ±0.00	100.00 ±0.00	±11.55

A. Result Analysis of CICIOMT2024 Dataset

The integration of SHAP-based feature selection with the Vision Transformer (ViT) architecture produces the most reliable results for IoMT ARP attack detection, as shown in Table VI. Specifically, the SHAP-ViT configuration achieves an Attack F1-score of 76% and a Balanced Accuracy of 84%, far outperforming baseline models trained on all attributes. Additionally, this technique significantly lowers the low recall rates observed in other architectures by raising the attack recall to 77% while maintaining strong discriminative power with an AUC of 0.87 and an AUPRC of 0.80. Table VII comparative results show that the SHAP-ordered ViT model works better than other setups for reconnaissance scenarios, especially when it comes to capturing attack instances where unordered techniques falter. Spatial feature orchestration is essential for optimizing transformer-based intrusion detection, as demonstrated by the SHAP-based methodology's significant recall improvement (87% versus 65% for the "All features" baseline). Table V gives an evaluation with respect to previous works performed on the same dataset. Fig. 2 displays the plot related to the balanced accuracy achieved for the models

B. Result Analysis of BoT-IoT Dataset

Table IX's comparative analysis demonstrates the ViT architecture's strong detection capabilities in OS fingerprinting situations. For the ViT model, both the SHAP and Correlation Coefficient-guided procedures obtain near-perfect or perfect attack-specific detection metrics; the Correlation Coefficient Ordered and SHAP-guided (ordered and unordered) approaches attain a 100% Attack F1-score. Table X demonstrates that the ViT model achieves 100% Attack F1-score, Recall, and Precision for BoT-IoT service scan assaults both under the baseline "All features" setup and employing the SHAP Ordered approach. These results demonstrate how feature organization techniques, especially SHAP-guided techniques, efficiently optimize the Vision Transformer's ability to identify scan-based reconnaissance. Table VIII performs a comparison of the proposed work with existing ones done with the BoT-IoT dataset. Fig. 3 displays the plot related to the balanced accuracy achieved for the models.

C. Discussion

Machine learning and deep learning models are generally trained and tested on the same attacks, and they don't have generalization capability, especially supervised models. In this work, we have used pretrained CNNs, InceptionNet, and ViT, which were trained only on DoS attacks and normal traffic and tested on different attacks, such as ARP and reconnaissance attacks. The network traffic is organised as an image and given as input to the models. Five cases are considered to compare the results: images synthesized from all features, features selected based on the correlation coefficient, and SHAP. Further features selected from the correlation coefficient and SHAP are taken as an unordered, functionality-based arrangement. It can be understood that not just selecting the features plays a role in improving noise filtering or improving computational efficiency, but their positioning further helps the models in understanding the spatial correlation existing among the features, thereby preventing the model from overfitting to specific attack noise by forcing it to learn the basic architecture of malicious traffic. Even if the particular "unknown" attack does not precisely match a DoS signature, the ViT can use its self-attention mechanism to focus on consistent, high-importance traffic anomalies that distinguish malicious intent from benign behavior when these primary-attack features are arranged structurally within the 2D image. When it comes to capturing local spatial connections, the Inception architecture excels. The InceptionNet is successfully presented with a "canonical" representation of a network anomaly when features are ordered functionally. This gives the model a consistent contextual anchor that helps it distinguish variants that it was not specifically trained to recognize. The SHAP-guided feature arrangement outperforms the 'All features' baseline in both attack types, with the ViT model's recall rising from 19% to 77% for ARP attacks and from 65% to 87% for reconnaissance attacks in the CICIoMT2024 dataset. It also achieves notable improvements in Balanced Accuracy and F1-score for both the Inception Network and ViT architectures. For BoT-IoT dataset, the SHAP-ordered feature arrangement consistently improves detection performance in both OS fingerprinting and service scan attack scenarios. Specifically, it increases the InceptionNet model's balanced accuracy from 65% to 99% for OS fingerprinting. At the same time, it allows the ViT model to reach peak 100% metrics in recall and F1-score for attack detection

D. Impact of Feature Selection and Spatial Ordering

When analyzing previously encountered attack types, feature subset selection (especially via SHAP) consistently improves attack resilience over the 'All features' baseline by effectively removing background noise. Additionally, by organizing similar network features into nearby neighborhoods, functional spatial layout plays a critical role. This spatial ordering significantly improves performance for the Vision Transformer (ViT). In difficult scenarios such as the CICIoMT2024 ARP Spoofing attacks, SHAP-guided ordering increased ViT's balanced accuracy by approximately 17% (from 67.20% to 78.60%) and its F1-score from 48.60% to 68.40%. Similarly, for Reconnaissance traffic, ViT's balanced accuracy improved to 87.60% and F1-score to 91.00%. This performance gain occurs because global self-attention is largely dependent on structured inter-feature layouts. Although its sliding-kernel and pooling mechanisms produce a more localized and varied response to overall spatial positioning, InceptionNet also benefits from functionality-based grouping; for instance, on the BoT-IoT OS Fingerprint baseline, InceptionNet's balanced accuracy surged by over 54% (from 64.00% to 98.60%), and its F1-score rose from 86.80% to 99.40% when transitioning from raw features to SHAP-ordered configurations.

E. Model Evaluation Across Random Seeds

Each of the experiments was carried out over five distinct random seeds (0, 7, 42, 123, 2026) in order to accurately assess the classification framework's stability, robustness, and reproducibility. In order to ensure that the reported metrics accurately reflect model performance rather than lucky weight initializations, this multi-seed evaluation protocol evaluates performance variance across both the baseline "All Features" configuration and the SHAP-ordered feature arrangement for the InceptionNet and Vision Transformer (ViT) architectures. For the CICIoMT2024 dataset, for recall metrics on ARP spoofing, where standard deviations can reach $\pm 24.08\%$ in ViT, the baseline configurations highlight that complex features introduce noticeable seed-dependent variance, underscoring the need for structured feature ordering (such as SHAP alignment) to stabilize convergence. Adopting the SHAP-ordered arrangement significantly improves performance and stabilizes consistency, increasing balanced accuracy to $98.60 \pm 0.55\%$ and recall to $98.80 \pm 0.45\%$, while the All-Features baseline configuration for InceptionNet on OS Fingerprint detection suffers from high seed instability ($64.00 \pm 2.45\%$ balanced accuracy and $\pm 32.33\%$ recall variance). While the All-Features baseline for service scan detection using InceptionNet already achieves a respectable balanced accuracy of $85.60 \pm 6.84\%$, using the SHAP-ordered feature arrangement significantly reduces variance and stabilizes performance, resulting in a more consistent balanced accuracy of $85.40 \pm 0.55\%$ with zero recall fluctuation ($\pm 0.00\%$). Table XI and Table XII show the performance across different metrics using random seed evaluation for both datasets.

V. Conclusion

Model generalization is an important challenge, particularly when it comes to network attack detection, since the attacker may launch a known attack differently. Therefore, the deep learning models should be ready for both known and unknown threats (related or unrelated to the attacks on which the model was trained). This study tries to enhance the performance of the model by utilizing a feature-orchestrated image synthesis technique on two benchmark datasets, BoT-IoT and CICIoMT2024, which include DoS, reconnaissance, and ARP spoofing attacks. FOS used the XAI technique, SHAP, a better model than LIME for performing local as well as global interpretation. Feature importance-based on DoS attacks and normal traffic is calculated, and the most relevant features are taken. These feature values are further scaled and tiled across a row, and the process is repeated for all feature values, thereby guiding the image synthesis process. These images are given to InceptionNet and ViT for the detection process. The

performance of the models is compared to image synthesis performed without any feature selection, as well as image synthesis using the correlation coefficient. In image synthesis using the correlation coefficient, selected features were correlated to DoS attacks. A balanced accuracy improvement of 25.86% for InceptionNet (from 59.00% to 74.20%) and 47.37% for ViT (from 67.20% to 99.00%, or relative gain calculations mapped from baseline to the structured configuration) is obtained for ARP Spoofing attack detection by implementing the SHAP-ordered feature arrangement over the baseline. The SHAP-ordered strategy consistently optimizes with a high F1 score of $87.00 \pm 0.00\%$ for service scan attack detection using InceptionNet, while ViT consistently performs well throughout assessment runs. It can be understood that feature-orchestrated input has a role in improving the unseen attack robustness of a model, especially for a supervised one. Future work can include the use of semi-supervised models to help the supervised model improve its generalisation capability

Conflict of Interest Statement

The authors declare no competing financial or personal interests that could have appeared to influence the work reported in this study.

References

[1] R. Morshedi and S. M. Matinkhah, "A comprehensive review of deep learning techniques for anomaly detection in IoT networks: Methods, challenges, and datasets," *Engineering Reports*, vol. 7, no. 9, p. e70415, 2025.

[2] S. K. R. Mallidi and R. R. Ramisetty, "Optimizing intrusion detection for IoT: a systematic review of machine learning and deep learning approaches with feature selection and data balancing," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 15, no. 2, p. e70008, 2025.

[3] Santosh K. Smmarwar, Govind P. Gupta, and Sanjay Kumar, "AI empowered malware detection system for industrial internet of things," *Computers and Electrical Engineering*, vol. 108, p. 108731, 2023.

[4] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 2027-2051, 2016.

[5] K. Saranya and A. Valarmathi, "A multilayer deep autoencoder approach for cross layer IoT attack detection using deep learning algorithms," *Scientific Reports*, vol. 15, no. 1, p. 10246, 2025.

[6] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, "A Comparative Analysis of Supervised Machine Learning Algorithms for IoT Attack Detection and Classification," *Journal of Computing Theories and Applications*, vol. 2, no. 3, pp. 395-409, 2025.

[7] G. Uthradevi, P. Thiruvasagam, S. Mythili, and S. O. Manoj, "A Semi-Supervised Deep Learning Approach for Intrusion Detection and Classification for the Internet of Things," *Biomedical Materials & Devices*, pp. 1-17, 2025.

[8] M. Al-Zewairi, S. Almajali, M. Ayyash, M. Rahouti, F. Martinez, and N. Quadar, "Multi-Stage Enhanced Zero Trust Intrusion Detection System for Unknown Attack Detection in Internet of Things and Traditional Networks," *ACM Transactions on Privacy and Security*, 2025.

[9] H. Hakami, M. Faheem, and M. B. Ahmad, "Machine learning techniques for enhanced intrusion detection in IoT security," *IEEE Access*, 2025.

[10] S. S. Alqahtany, A. Shaikh, and A. Alqazzaz, "Enhanced Grey Wolf Optimization (EGWO) and random forest based mechanism for intrusion detection in IoT networks," *Scientific Reports*, vol. 15, no. 1, p. 1916, 2025.

[11] U. C. Akuthota and L. Bhargava, "Transformer-based intrusion detection for IoT networks," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 6062-6067, 2025.

[12] M. Benmalek and A. Seddiki, "Particle swarm optimization-enhanced machine learning and deep learning techniques for Internet of Things intrusion detection," *Data Science and Management*, 2025.

[13] D. D. Bikila and J. Čapek, "Machine Learning-Based Attack Detection for the Internet of Things," *Future Generation Computer Systems*, vol. 166, p. 107630, 2025.

[14] M. R. Minhas et al., "F-OSFA: A Fog Level Generalizable Solution For Zero-day DDOS Attacks Detection," *IEEE Access*, 2025.

[15] Santosh K. Smmarwar, Govind P. Gupta, and Sanjay Kumar, "AI empowered malware detection system for industrial internet of things," *Computers and Electrical Engineering*, vol. 108, p. 108731, 2023.

[16] X. H. Nguyen and K. H. Le, "Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model," *Internet of Things*, vol. 23, p. 100851, 2023.

[17] Belal Ibrahim Hairab et al., "Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks," *IEEE Access*, vol. 10, pp. 98427-98440, 2022.

[18] Segun I. Popoola et al., "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930-3944, 2021.

[19] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. Ghorbani, "Ciciomt2024: Attack vectors in healthcare devices-a multi-protocol dataset for assessing iomt device security," 2024.

[20] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019.

[21] P. Chen, F. Li, and C. Wu, "Research on intrusion detection method based on Pearson correlation coefficient feature selection algorithm," *Journal of Physics: Conference Series*, vol. 1757, no. 1, p. 012054, 2021.

[22] U. Ahmed, Z. Jiangbin, A. Almogren, M. Sadiq, A. U. Rehman, M. T. Sadiq, and J. Choi, "Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems," *Scientific Reports*, vol. 14, no. 1, p. 30532, 2024.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1--9, 2015.

[25] lfageer, R., Ghaleb, M., Aljoby, W., \& Felemban, M. (2026). A Multi-Stage Learning Framework for Detecting Known and Novel Attacks in the Internet of Medical Things. IEEE Access, 14, 48312-48326.

[26] Torre, D., Chennamaneni, A., Jo, J., Vyas, G., \& Sabrsula, B. (2025). Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study. ACM Transactions on Software Engineering and Methodology, 34(2), 1-48.

[27] Jodayree, M., Ghafi, A. K., Amiri, S., \& Shaykholeslami, P. (2026). Explainable zero-day attack detection in IoMT using transformer-based time-series modeling. Scientific Reports.

[28] Gueriani, A., Kheddar, H., Mazari, A. C., \& Ghanem, M. C. (2025). A robust cross-domain IDS using BiGRU-LSTM-attention for medical and industrial IoT security. ICT Express.

[29] D. Torre, A. Chennamaneni, J. Jo, G. Vyas, and B. Sabrsula, "Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 2, pp. 1--48, 2025.

[30] M. Luqman, M. Zeeshan, Q. Riaz, M. Hussain, H. Tahir, N. Mazhar, and M. S. Khan, "Intelligent parameter-based in-network IDS for IoT using UNSW-NB15 and BoT-IoT datasets," *Journal of the Franklin Institute*, vol. 362, no. 1, p. 107440, 2025.

Arabic Text Simplification and Readability Assessment for Educational Applications: A Systematic Literature Review

Khalid Essaadani, Soumia Ziti
Mohammed V University in Rabat, Morocco

***Abstract*—Arabic text simplification and readability assessment support level-appropriate material selection, language learning, early literacy, teacher adaptation, and accessibility, but the degree of educational validation remains uncertain. This PRISMA 2020 systematic review searched Scopus, Web of Science Core Collection, and IEEE Xplore on 6 August 2026 and updated coverage through a targeted ACL Anthology search, a standalone-readability sensitivity search, and citation-cluster checking on 11 September 2026. Following source-identity verification, 194 records were identified, 129 were screened, 69 full texts were assessed, and 61 studies were included. An eight-criterion appraisal rated 45 studies high, 13 moderate, and 3 low quality. Thirty-four studies primarily modelled readability, while recent work increasingly used Arabic pretrained language models, ordinal objectives, graph models, large corpora, and controlled generation. Forty-two studies relied primarily on automatic or benchmark metrics; only two directly measured learner comprehension or retelling, and none measured longitudinal learning gain. Excluding low-quality studies and interpreting documented resource reuse did not alter the central conclusion. The retrieved evidence demonstrates technical feasibility and growing educationally relevant resources, but educational effectiveness remains unestablished. Priorities are independent learner and teacher evaluation, cross-domain and cross-variety validation, interoperable scales, human calibration of automatic metrics, and closed-loop diagnosis, simplification, and level verification.**

Keywords*—Arabic text simplification; Arabic readability assessment; educational natural language processing; systematic literature review

I. Introduction

Matching a text to its reader is a central educational requirement. Readability assessment estimates text difficulty or predicts an appropriate reader level; text simplification modifies lexical, syntactic, or discourse complexity while attempting to preserve meaning. In educational systems, the tasks are complementary: assessment can identify a mismatch, simplification can adapt the material, and reassessment can test whether the target level was reached.

Arabic complicates this cycle through rich morphology, optional diacritics, orthographic variation, lexical-frequency effects, and diglossia. Modern Standard Arabic (MSA), Classical Arabic, and spoken varieties place different demands on readers. The relevant populations range from first-language (L1) school readers to Arabic second- or foreign-language learners (L2), university students, emergent readers, teachers, and people who need accessible text.

The central evidence problem is not whether a model improves a benchmark score, but whether the improvement transfers to a defined reader and produces a meaningful outcome. Model accuracy, annotation agreement, and linguistic judgments are necessary forms of evidence, yet they do not by themselves establish comprehension, retention, learning, teacher adoption, or sustained classroom benefit.

A. Background and Prior Reviews

Arabic readability research includes corpus-frequency formulas [3], feature-rich machine learning, leveled corpora, and neural models. Mat Daud et al. propose a formula based on reversed word-frequency ranks, while discussing length-based formulas as prior work [3]. Nassiri, Cavalli-Sforza, and Lakhouaja's survey [1] describes formula-based and machine-learning approaches to Arabic readability for L1, L2, and adult audiences, according to its published abstract.

Building on that readability-focused account, the present review jointly examines readability assessment and text simplification under an educational scope. Its contribution is an explicit connection between study-level technical findings and the kind of educational evidence they provide: resource validity, reader processing, usability, direct learner outcomes, or learning gain. PRISMA selection, criterion-level appraisal, the September 2026 search update, and documented evidence-family mapping make this connection auditable. The earlier survey's full text was unavailable for this revision; this comparison is therefore limited to its published scope and does not assert that particular procedures or topics were absent from it.

Its contributions are a combined synthesis of Arabic text simplification and readability assessment under an explicit educational scope; a reproducible search, deduplication, screening, extraction, and appraisal trail; a taxonomy of educational contexts, computational approaches, resources, and evaluation practices; an evidence-level distinction between technical performance, reader validity, and educational effectiveness; and a prioritized research agenda for learner-centred, teacher-centred, and classroom evaluation.

B. Research Questions

- RQ1: Which educational objectives, learner populations, and application contexts are addressed?

- RQ2: Which computational or analytical approaches, tools, and techniques are used?
- RQ3: Which resources, datasets, annotation schemes, and benchmarks support the field?
- RQ4: How are systems and resources evaluated, and what educational outcomes are measured?
- RQ5: Which evidence gaps and research priorities emerge across the literature?

II. Review Method

A. Review Design and Scope

The review followed PRISMA 2020 [2]. The protocol was developed before the original screening but was not prospectively registered. The unit of inclusion was a peer-reviewed primary journal article or full conference paper examining Arabic text simplification, readability, reading level, text difficulty, or text complexity with an explicit educational, learner-facing, teacher-facing, literacy, or accessibility connection.

Table I states the operational eligibility rules. Retrieval failure was recorded separately and was not treated as a content-based exclusion.

B. Information Sources and Search Strategy

Scopus, the Web of Science Core Collection, and IEEE Xplore were searched on 6 August 2026. Each English query combined Arabic identifiers with phrases for text simplification, readability assessment, reading level, text difficulty, and text complexity. Arabic-script supplementary queries returned zero records. Journal articles and full conference papers were retained through database-specific type filters; no year, country, subject, document-language, or access-status restriction was used.

To test the NLP coverage gap identified during review, the official ACL Anthology title-and-abstract metadata were searched on 11 September 2026 with the original concept blocks. A sensitivity query added the standalone term "readability," which had been omitted originally to limit unrelated uses. The main ACL query returned 25 candidates, and the standalone-readability query returned 16 non-overlapping candidates. Five additional BAREC system papers were located by citation-cluster checking because their searchable metadata did not explicitly repeat an Arabic identifier. After removal of five overlaps with the original set and full-text screening of 41 unique reports, seven were excluded, and 34 were newly included. The standalone-readability search contributed 10 of these inclusions, demonstrating that the term materially improved recall.

TABLE I. Eligibility Criteria and Operational Exclusions

Criterion	Include	Exclude or record separately
Language focus	Arabic is the target language or Arabic results are separable.	Non-Arabic work, Arabic-script work in another language, or multilingual work without separable Arabic evidence.
Target phenomenon	Text simplification, readability assessment/prediction, reading level, text difficulty, or text complexity.	Adjacent tasks in which readability is only incidental.
Educational relevance	Learners, teachers, curricula, material selection, literacy, accessibility, or another explicit learner-facing use.	Domain readability with no defensible educational or learner-facing connection.
Evidence type	Original computational, corpus, system, usability, reader, learner, or pedagogical research.	Editorials, narrative surveys, perspectives, and reviews.
Publication type	Peer-reviewed journal article or full conference paper.	Thesis, abstract-only record, poster, book/chapter, standard, or non-peer-reviewed report.
Assessment	Sufficient full text to judge eligibility and extract evidence.	Unavailable reports were recorded as not retrieved; they were not assigned a substantive exclusion reason.

ACM Digital Library coverage was checked through ACM DOI-prefix metadata because the public interface triggered security verification and did not provide a reproducible export during the revision. Two ranked searches ("Arabic readability" and "Arabic text simplification") identified no unique eligible primary report: two potentially eligible reports were already in the database set, and two Arabic-focused records were reviews. Exact queries, dates, filters, counts, clean copyable Arabic-script strings, and record-level decisions are supplied in the supplementary workbook.

C. Record Management, Retrieval, and Selection

Records were deduplicated first by normalized DOI and then by normalized title, while retaining provenance. Title and abstract records were coded Include, Maybe, or Exclude with a reason; Include and Maybe records advanced to retrieval. Additional attempts for the original 15 unavailable reports comprised publisher/DOI checks, exact-title web and institutional-repository searches, ResearchGate availability checks, and OpenAlex open-access-location verification. A source-identity audit on 16 September 2026 found that the PDF stored for ATSRA-010 [9] was a different publication by the same authors. Publisher and exact-title checks did not yield the intended full text; the linked ResearchGate PDF repeated the mismatch. ATSRA-010 was therefore reclassified as not retrieved, and its extraction and appraisal were withdrawn. The final total is 16 unretrieved reports. This was a retrieval correction, not a content-based exclusion or a new search inclusion. The reports, attempts, and topic-only coverage implications are listed in the supplement. No result, direction, or effect was imputed from an unretrieved report.

D. Data Extraction and Evidence Dependence

For every included report, the extraction table records citation, year, setting, population or educational context, Arabic variety, task, method, dataset and size, readability scale, evaluation design and metrics, principal findings, limitations, and evidence relevant to RQ1–RQ5. Publications were additionally mapped to evidence families. BAREC corpus, guidelines, demo, shared-task, and system papers form a shared

benchmark family; SAMER, DARES/SaudiTextBooks, AraEyebility, ATSC, and the L2/GLOSS research programme also contain repeated or evolving data. Documented reuse is distinguished from related research with unverified participant overlap, including the earlier 2021 eye-tracking pilot. These mappings guide interpretation but do not establish an exact count of independent datasets or participant samples.

E. Quality Appraisal

Eight equally weighted criteria were scored 0, 0.5, or 1: clear aims and educational relevance; description of data/sample/context/Arabic variety; reproducibility; claim-appropriate evaluation; design-appropriate baseline, control, agreement, or validation; clear results and suitable metrics; proximity to intended readers, learners, educators, or users; and limitations/transparency/availability. Criterion-specific anchors are provided in the supplement. For example, Q5 accepts competitive baselines or ablations in computational studies, inter-annotator agreement or held-out benchmarks in corpus studies, and comparison/control evidence in user studies. Q7 deliberately records educational-evidence proximity: direct comprehension, retelling, retention, learning gain, or authentic educator adoption scores 1; indirect human/expert/user evidence scores 0.5; no human evidence scores 0.

Totals of 6.5–8 were classified high (at least 81.25% of the maximum), 4.5–6 moderate (56.25–75%), and 0–4 low (50% or less). These pragmatic thresholds distinguish broadly complete, partially complete, and substantially limited evidence profiles. Quality was not an exclusion rule. High-quality technical studies could strongly support a model or resource claim while remaining weak evidence of educational effectiveness.

F. Review Procedure and Verification

The first reviewer conducted the full screening, data extraction, and quality appraisal. The coauthor then independently re-screened a blinded, stratified sample of 13 of 129 records (10.1%) using the same Include, Maybe, or Exclude categories and eligibility criteria. Exact three-category agreement was 7/13 (53.8%; Cohen's kappa = 0.284). Because both Include and Maybe advanced to retrieval, agreement on the operational advance-versus-exclude decision was 11/13 (84.6%; kappa = 0.683). Four differences were conservative Include-versus-Maybe judgments and did not alter progression. The first reviewer adjudicated all differences against the prespecified criteria and available full texts. ATSRA-017 was changed from Exclude to Include because the protocol explicitly includes accessibility and learner-facing information; SUP-001 remained excluded because text difficulty was used only as a machine-translation performance variable. Extraction and quality appraisal were not independently duplicated and remain a limitation. The screening sample, reasons, agreement calculations, adjudications, complete extraction data, and item-level ratings are supplied in the supplementary materials.

G. Synthesis

A narrative thematic synthesis was organized around RQ1–RQ5. Meta-analysis was inappropriate because populations, tasks, scales, datasets, and outcomes were heterogeneous. Four categories were used: direct learner outcome (comprehension, Cloze, retelling, retention, or achievement in intended learners); reader-processing evidence (eye movements, time, or perceived difficulty without a learning-gain design); usability evidence (workflow or acceptability without comprehension); and learning gain (pre/post or longitudinal improvement attributable to an intervention). Automatic metrics, annotation reliability, expert linguistic judgments, corpus statistics, and demonstrations were classified as technical/resource evidence.

III. Results

A. Study Selection

The original databases yielded 148 records. Supplementary ACL searching and citation-cluster checking identified 46 additional records, for 194 in total. Removal of 65 duplicates left 129 records for screening; 44 were excluded at title and abstract screening. Eighty-five reports were sought, 16 were not retrieved after source-identity verification, and 69 were assessed for eligibility. Eight reports were excluded after full-text assessment (seven supplementary reports outside the review scope and one master's thesis), leaving 61 included studies (Fig. 1) (Table V). The 16 unretrieved reports represented 36.4% of the 44 distinct reports sought through the original database pathway and 18.8% of all 85 reports sought after the update.

B. Study Characteristics and Evidence Structure

The 61 publications appeared from 2013 to 2026; 48 (78.7%) were published from 2022 onward. Table II summarizes the evidence base and distinguishes publication volume from evidential independence. Readability modelling was the largest primary task (n = 34), followed by resource/corpus work and simplification or readability-controlled generation (n = 8 each), tools/systems (n = 6), and human/pedagogical assessment (n = 5). The concentration of 2025 publications partly reflects 18 BAREC shared-task and overview papers that evaluate a common benchmark.

TABLE II. Distribution of the Included Evidence Base

Dimension	Distribution	Total
Quality tier	High 45; moderate 13; low 3	61
Primary task	Readability modelling 34; resource/corpus 8; simplification or controlled generation 8; tool/system 6; human/pedagogical assessment 5	61
Primary evaluation	Automatic/benchmark metrics 42; resource quality 6; system/demo 5; learner/reader experiment 3; expert simplification evaluation 2; usability 1; pedagogical analysis 1; prototype 1	61
Direct educational evidence	Direct learner outcomes 2; learning-gain studies 0	2 direct; 0 gain
Major dependent families	BAREC; SAMER; DARES/SaudiTextBooks; AraEyebility; ATSC; L2/GLOSS	6 mapped families

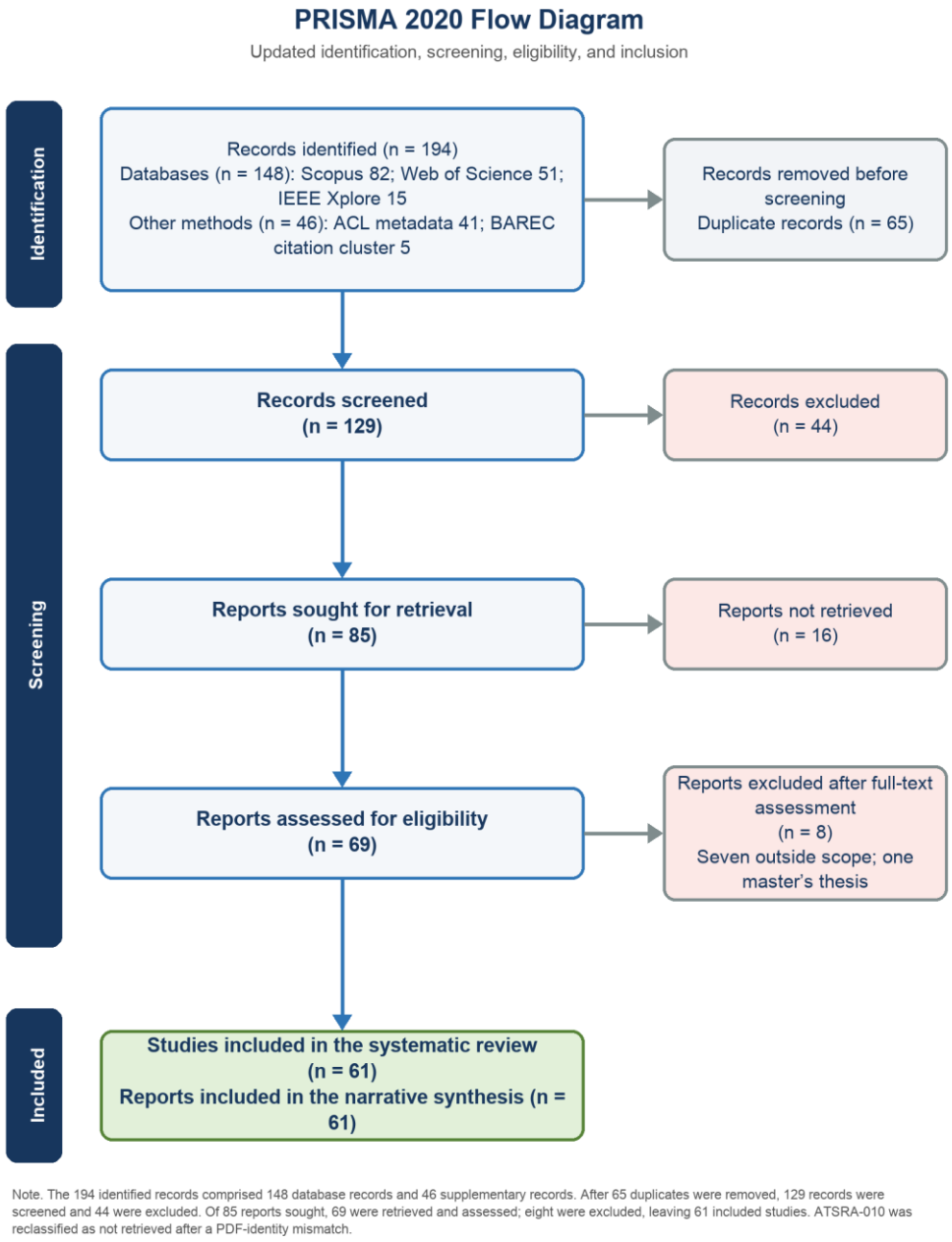


Fig. 1. PRISMA 2020 flow diagram for the updated review.

C. *Quality Appraisal and Quality-Informed Synthesis*

Forty-five studies were high quality, 13 moderate, and three low; the mean was 6.45 of 8. The item-level table shows a recurrent pattern: data, methods, and benchmark results are usually well reported, while Q7 is often zero because intended readers or educators are not evaluated. This does not invalidate technical findings; it limits the claim they can support.

Quality weighting was applied concretely. High-quality BAREC, SAMER, DARES, and cross-domain studies anchor claims about corpus maturity, model performance, and domain sensitivity [4], [15], [29], [37]–[44], [55]–[63]. Moderate-quality demos and accessibility audits describe workflows or text properties but do not establish learner benefit [10], [12], [30], [34], [64]. The three low-quality reports, including the CART prototype [13], contribute historical or design context only and do not anchor any conclusion. Omitting them leaves 58 studies; both direct learner studies remain, no learning-gain study appears, and the conclusion of technical feasibility without established educational effectiveness is unchanged (Table III).

TABLE III. QUALITY-INFORMED SENSITIVITY ANALYSIS

Analysis	n	High/Moderate/Low	Direct learner outcomes	Learning gain	Interpretation
All included	61	45/13/3	2	0	Technical feasibility; effectiveness unestablished
Exclude low quality	58	45/13/0	2	0	Central conclusion unchanged
Account for evidence families	Publication total retained; dependent resources mapped	Family-level interpretation	2 independent learner samples	0	Model volume remains concentrated around shared resources

D. RQ1: Educational Objectives, Learners, and Applications

The literature proposes four principal educational uses. First, readability prediction supports material selection, curriculum grading, and passage matching for school, university, and L2 readers [7], [8], [10], [15], [22], [23], [29], [32], [35], [37], [42], [58]. Second, simplification systems and resources support lexical substitution, sentence rewriting, accessible fiction, religious text adaptation, and readability-controlled generation [5], [6], [10], [25], [29], [36], [39], [45], [56], [57]. Third, tools expose difficulty information to teachers, authors, annotators, and learners through reports, visualization, thesauri, and searchable corpora [10], [12], [16], [30], [34]. Fourth, reader studies examine textbook suitability, early literacy, variety, and cognitive processing [7], [11], [14], [18], [20], [27].

The demonstrated educational impact is much narrower than the proposed use. In the university Cloze study, 54 students completed four unseen textbook passages; passage mean scores ranged from 0.46 to 0.66, and all four passages were classified at the frustration level for that sample [7]. The study directly tests material–learner fit but is limited to one institution and textbook. In the preschool experiment, 30 Arabic-speaking children (mean age 5 years 10 months) encountered four picture-supported narratives that varied in MSA versus spoken Palestinian Arabic and in episodic complexity. Comprehension was higher for MSA, while greater episodic complexity supported longer retellings; these are outcome-specific findings, not uniform effects across all measures [27]. Its direct child evidence is valuable but limited by a small, relatively advantaged sample and four narratives.

These two reports are the only direct learner-outcome studies. The three studies counted as primary learner/reader experiments comprise those two plus one CA–MSA eye-tracking comparison of 15 matched proficient adults: 23 were recruited, 18 retained for MSA, and 15 completed both sessions [18]. Other eye-tracking papers use gaze data primarily for resource construction or prediction [11], [14], [20]. The usability study involved five graduate human–computer interaction students evaluating four existing general readability tools to inform Arabic reporting interfaces, not an Arabic learner intervention [16]. Thus, "three learner/reader experiments" and "two direct learner outcomes" describe different evidence categories and are not conflicting counts. No study measured longitudinal learning gain.

E. RQ2: Methods, Tools, and Techniques

Formula and rule-based work uses surface length, syllable structure, lexical density, word frequency, diacritics, and Arabic-specific factors such as Faseeh. The corpus-frequency formula in [3] draws on KACSTAC (739,119,011 words); this is its source corpus, not a learner-validation sample. OSMAN evaluated an Arabic formula against English readability scores on approximately 73,000 aligned UN text units, described as sentences in its abstract and paragraphs in its methods [31]. These methods are transparent and inexpensive, but formula correlation is not equivalent to learner calibration.

Classical machine learning uses linguistic features with support-vector methods and random forests [17], [21]. SATLab instead combines character n-grams with support-vector regression and optimized ordinal thresholds, providing a non-neural BAREC baseline [60]. Feature optimization reported 94.8% L1 accuracy and 72.4% L2 SVM accuracy in its main setting; its augmented-feature experiment additionally reached 74.1% L2 accuracy with kNN (Table VII in [32]). On the Jordanian School Textbook Corpus, CAMeLBERT reached in-domain QWK 0.830 but fell to 0.085 cross-domain, whereas a 127-feature XGBoost model retained the best cross-domain QWK of 0.240 [58].

Neural approaches include fastText and contextual embeddings, CNN/BiLSTM architectures, Arabic pretrained transformers, graph neural networks, ordinal heads, differentiable QWK losses, conformal decoding, ensembles, and visual text models [19], [29], [35], [38], [40]–[43], [46]–[54], [59], [61]–[63]. In the BAREC shared task, the strongest system reached QWK 0.875 at sentence level and 0.874 at document level [40], [42]. Because 18 reports share BAREC data and tracks, these results compare techniques within one family rather than supplying 18 independent validations.

Simplification methods range from lexical pipelines and embedding-based substitutions to Seq2Seq, mT5, switch transformers, machine-translation hybrids, and prompted LLM generation [5], [6], [25], [29], [36], [39], [45], [56], [57]. The 2023 embedding system reached complex-word identification F1 of 0.88; expert review found that 79% of binary-model outputs were simpler, but the multi-complexity model better preserved meaning (69%) and grammar (84%), demonstrating a real simplification–adequacy trade-off [36]. Controlled CEFR prompting reached high automatic profile alignment [56], but still lacked learner testing.

Human-centred techniques include Cloze testing, narrative comprehension and retelling, expert linguistic judgments, usability observation, and eye tracking. The later AraEyebility corpus and analyses reuse related data [11], [18], [20]. The earlier 41-participant pilot [14] belongs to the same research programme, but participant overlap with the later corpus is unverified. Small samples and documented reuse limit generalization; a shared research programme alone does not establish identical participants.

F. RQ3: Resources, Datasets, and Scales

Resource capacity has expanded substantially. BAREC contains 69,441 sentences and more than one million words across 19 levels, with QWK 0.818 in its final annotation phase [4], [44]; BAREC-10M extends the programme to approximately 10 million words and 45 subcorpora [55]. SAMER contains approximately 159,000 original-corpus words from 15 novels with word/document levels and two simplified versions [39], and its associated lexicon contains approximately 26,000 triply annotated lemmas [33]. DARES 1.0 includes 13,335 curriculum instances [37], while DARES 2.0 expands and cleans concept-centred Saudi textbook data [15]. Other resources include ATSC (500 article/simplification pairs) [6], README++ (9,757 multilingual sentences, including 1,945 Arabic) [24], AraEyebility [11], MoSAR [23], Saaq al-Bambuu (2,980 sentence pairs) [29], and the 82,512-segment, 240-textbook JSTC [58].

Interoperability remains weak. Taha/Arabi21 levels, school grades, the Common European Framework of Reference, Interagency Language Roundtable levels, Cloze thresholds, and bespoke labels differ in target population and unit of analysis. Dataset expansion therefore does not automatically produce transferable educational meaning. Crosswalks should report uncertainty and be validated with intended readers.

G. RQ4: Evaluation and Educational Outcomes

Forty-two publications primarily report automatic or benchmark metrics such as accuracy, macro/weighted F1, QWK, BLEU, SARI, METEOR, ROUGE, BERTScore, or distance from an ordinal label. Six focus on resource quality and five on system or demo functionality. These evaluations can establish technical reproducibility, annotation reliability, or relative performance, but not comprehension or learning.

Evidence categories are exclusive at the primary-evaluation level but are interpreted alongside secondary human evidence. Direct learner outcome requires task performance by intended learners; reader processing requires cognitive or behavioral reading measures without a learning-gain design; usability concerns workflow and acceptability; learning gain requires change attributable to an intervention. The observed counts are two, three, one, and zero, respectively. Expert judgments of grammar, meaning, or annotation quality strengthen linguistic validity but are not learner outcomes.

H. RQ5: Gaps and Priorities

The dominant gap is educational validation: 59 of 61 publications lack direct learner outcomes, and no longitudinal classroom intervention was found. Other priorities are teacher-centred evaluation, external and cross-domain testing, dialect/register coverage, calibration of automatic metrics against Arabic comprehension, transparent cross-scale mapping, and independent replication beyond recurring datasets. Unretrieved reports appear to concern accessibility, early tools, feature/vectorization methods, L2 readability, class balancing, patient education, lexical simplification, and web indices; their absence may narrow historical and applied coverage, but no unobserved result is assigned to them (Table IV).

TABLE IV. EVIDENCE-BASED RESEARCH AGENDA

Priority	Evidence gap	Recommended design	Required outcomes
Critical	Direct learner evidence remains 2/61; learning-gain evidence is 0/61.	Controlled or quasi-experimental studies with intended learners.	Comprehension, retention, reading efficiency, learning gain.
Critical	No longitudinal classroom deployment.	Teacher co-design and sustained authentic-course deployment.	Adoption, workload, instructional decisions, sustainability.
High	Strong benchmark but weak cross-domain transfer.	External validation across curricula, genres, regions, and varieties.	QWK/F1 plus subgroup validity and error analysis.
High	Automatic metrics dominate 42 primary evaluations.	Calibrate metrics against Arabic human judgments and learner tasks.	Meaning, grammaticality, comprehensibility, comprehension.
High	Readability scales are not interoperable.	Multi-scale annotation and transparent probabilistic crosswalks.	Reliability, mapping uncertainty, transfer validity.
High	Publication volume clusters around shared resources.	Independent datasets, preregistered splits, and evidence-family reporting.	External replication and generalization.
High	Few closed-loop educational systems.	Diagnose difficulty, simplify, reassess target level, then test with users.	Target attainment, comprehension, teacher control.

IV. DISCUSSION

A. Principal Findings and Educational Applications

Research on Arabic readability and simplification is rapidly expanding. The updated search reveals a mature technical ecosystem of leveled lexicons, school and literary corpora, annotation tools, visual interfaces, feature-based classifiers, Arabic transformers, ordinal objectives, and controlled generators. It also strengthens—not weakens—the original central message: adding 34 ACL studies increases the amount of technical evidence without adding a direct learner-outcome or learning-gain study.

For material selection and curriculum design, practical evidence combines direct mismatch detection in the university Cloze study [7], child variety/complexity effects [27], school-corpus modelling [15], [37], [58], and textbook classification and gradation analysis [26]. For teacher authoring, tools can flag difficult words, suggest alternatives, expose levels, and search graded corpora [10], [12], [30], [34], while the five-person general-tool usability study offers interface-design guidance rather than educator-adoption evidence [16]. Adoption, workload, and decision quality remain insufficiently measured. For L2 learning, CEFR/ILR/GLOSS work shows feasible automatic leveling [8], [28], [32], [35], yet the labels require calibration to real learners.

For accessibility, the Arabic orthodontic-information audit covered 74 websites and found apparently accessible formula scores but limited reliability of the online information and no direct patient-comprehension test [64]. Other disability and patient-education reports remain unretrieved, so conclusions in these subdomains remain cautious.

Generative systems create a new educational opportunity and a new validity risk. Models can simplify, control grade or CEFR profiles, and diversify early-grade stories [5], [6], [25], [29], [36], [39], [45], [56], [57]. However, automatic level agreement can coexist with factual drift, loss of meaning, cultural mismatch, or inappropriate difficulty for a real learner. Educational deployment therefore requires human evaluation, reader testing, educator control, and post-generation reassessment.

B. Quality, Dependence, and Strength of Conclusions

The quality-informed synthesis prevents high benchmark scores from carrying inappropriate educational weight. High-quality computational studies anchor statements about what works on their datasets; direct learner reports anchor the much narrower statements about comprehension and text–reader fit. The low-quality reports are contextual only. Excluding them does not change the conclusion. Likewise, evidence-family mapping shows that the BAREC surge represents intensive comparison on one valuable benchmark, not many independent populations. Similar caution applies to AraEyebility participant subsets and successive DARES, SAMER, ATSC, and L2/GLOSS outputs.

C. Strengths and Limitations

Strengths include exact database queries, a targeted ACL update, a standalone-readability sensitivity search, citation-cluster checking, explicit record-level decisions, retrieval auditing, study-level extraction, criterion-anchored appraisal, sensitivity analysis, and evidence-family mapping. The supplementary workbook makes each main conclusion traceable to individual reports.

Limitations remain. Sixteen reports could not be retrieved, including one whose stored PDF proved to be a different publication; they represented 36.4% of reports sought through the original database pathway and 18.8% overall after the update. The source-identity error led to withdrawal of that report's extraction and appraisal. ACM coverage was checked indirectly through DOI-prefix metadata because the interface did not permit reproducible access during revision. The protocol was not registered prospectively. Heterogeneity precluded meta-analysis. Independent coauthor re-screening covered 10.1% of records and showed substantial agreement on the operational advance-versus-exclude decision, but only fair exact three-category agreement. Extraction and appraisal were not independently duplicated, so reviewer-dependent judgment may still have affected extracted classifications and quality scores. The comparison with the earlier survey is limited to its published abstract, and some participant-overlap relationships remain unverified. Finally, 2026 indexing is still evolving, and later records may alter counts.

V. Conclusion

Research on Arabic text simplification and readability assessment has progressed from formulas and handcrafted indicators to large, fine-grained corpora, cognitive signals, Arabic pretrained language models, graph and ordinal methods, and readability-controlled generation. These advances create credible tools for level-aware material selection, text adaptation, curriculum analysis, and educational content generation.

They do not yet establish broad educational effectiveness. Only two of 61 included publications directly measured learner outcomes, none measured longitudinal learning gain, and much of the recent model literature reuses a small number of resource families. The camera-ready conclusion is therefore bounded: technical feasibility and resource maturity are supported by the retrieved evidence; improved learning or classroom practice is not. The next phase should connect models to representative learners, teachers, curricula, varieties, and authentic use, while preserving meaning and independently verifying target-level attainment.

Supplementary Materials

The supplementary review data, including the database search log, screening decisions, unretrieved-report audit, extraction table, study-level evidence tables, evidence-family map, quality-appraisal ratings, sensitivity analyses, evidence-category definitions, and independent coauthor verification, are publicly available in Zenodo at https://zenodo.org/records/22829321, DOI: 10.5281/zenodo.22829321.

Declarations

Ethics approval: Not applicable. This review analyzed published literature and collected no new human-participant data.

Funding: This research received no external funding.

Conflicts of interest: The authors declare no conflict of interest.

Data availability: The review dataset and supporting supplementary materials are publicly available in Zenodo at https://zenodo.org/records/22829321, DOI: 10.5281/zenodo.22829321.

Declaration on generative AI: Generative AI tools assisted spreadsheet organization, document preparation, and language refinement. The authors remain responsible for eligibility decisions, evidence interpretation, citation verification, originality, and final content.

References

[1] N. Nassiri, V. Cavalli-Sforza, and A. Lakhouaja, "Approaches, methods, and resources for assessing the readability of Arabic texts," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 4, Art. no. 95, 2023. doi: 10.1145/3571510.

[2] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," BMJ, vol. 372, Art. no. n71, 2021. doi: 10.1136/bmj.n71.

[3] N. Mat Daud, H. Hassan, and N. Abdul Aziz, "A corpus-based readability formula for estimate of Arabic texts reading difficulty," World Applied Sciences Journal, vol. 21, no. SPECIAL ISSUE4, pp. 168–173, 2013.

[4] K. N. Elmadani, N. Habash, and H. Taha-Thomure, "A Large and Balanced Corpus for Fine-grained Arabic Readability Assessment," in Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 16376–16400. doi: 10.18653/v1/2025.findings-acl.842.

[5] A. Alsughayyir, and A. Alshanqiti, "A light but efficient switch transformer model for Arabic text simplification," International Journal of Advanced and Applied Sciences, vol. 12, no. 4, pp. 71–78, 2025. doi: 10.21833/ijaas.2025.04.009.

[6] A. Alshanqiti, A. Alkhodre, A. Namoun, S. Albouq, and E. Nabil, "A Transformer Seq2Seq Model with Fast Fourier Transform Layers for Rephrasing and Simplifying Complex Arabic Text," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, pp. 888–898, 2023. doi: 10.14569/IJACSA.2023.01402101.

[7] A.H.H. Ahmad, and Y.N. Hasan, "Applied Arabic Textbook Readability Level for University Students at Al-Balqa Applied University," Theory and Practice in Language Studies, vol. 14, no. 1, pp. 186–191, 2024. doi: 10.17507/tpls.1401.22.

[8] N. Nassiri, A. Lakhouaja, and V. Cavalli-Sforza, "Arabic L2 readability assessment: Dimensionality reduction study," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 3789–3799, 2022. doi: 10.1016/j.jksuci.2020.12.021.

[9] N. Nassiri, A. Lakhouaja, and V. Cavalli-Sforza, "Arabic readability assessment for foreign language learners," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, pp. 480–488. doi: 10.1007/978-3-319-91947-8_49.

[10] R. Hazim, H. Saddiki, B. Alhafni, M. Al Khalil, and N. Habash, "Arabic Word-level Readability Visualization for Assisted Text Simplification," in EMNLP 2022 - 2022 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Demonstrations Session, 2022, pp. 242–249. doi: 10.18653/v1/2022.emnlp-demos.24.

[11] I. Baazeem, H. Al-Khalifa, and A. Al-Salman, "AraEyebility: Eye-Tracking Data for Arabic Text Readability," Computation, vol. 13, no. 5, Art. no. 108, 2025. doi: 10.3390/computation13050108.

[12] K. Altarbouch, K.N. Elmadani, O. Obeid, H. Taha-Thomure, and N. Habash, "BAREC Demo: Resources and Tools for Sentence-level Arabic Readability Assessment," in EMNLP 2025 - 2025 Conference on Empirical Methods in Natural Language Processing, Proceedings of the System Demonstrations, 2025, pp. 950–959. doi: 10.18653/v1/2025.emnlp-demos.74.

[13] A. H. Ishraf Shuib, N. A. Aziz, N. M. Daud, and H. Hasan, "CART — An Arabic teaching-learning tool for non-native Arabic speakers," in The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), 2014, pp. 1–4. doi: 10.1109/ICT4M.2014.7020585.

[14] I. Baazeem, H. Al-Khalifa, and A. Al-Salman, "Cognitively driven arabic text readability assessment using eye-tracking," Applied Sciences (Switzerland), vol. 11, no. 18, Art. no. 8607, 2021. doi: 10.3390/app11188607.

[15] S. Almujaiwel, D. Premasiri, T. Ranasinghe, M. El-Haj, and R. Mitkov, "Complex Concept-Based Readability Estimation from Arabic Curriculum," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 24, no. 11, Art. no. 128, 2025. doi: 10.1145/3770070.

[16] H. Saddiki, V. Cavalli-Sforza, and K. Bouzoubaa, "Enhancing Visualization in Readability Reports for Arabic Texts," in Procedia Computer Science, 2017, pp. 241–247. doi: 10.1016/j.procs.2017.10.115.

[17] E. Salesky and W. Shen, "Exploiting Morphological, Grammatical, and Semantic Correlates for Improved Text Difficulty Assessment," in Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, 2014, pp. 155–162. doi: 10.3115/v1/W14-1819.

[18] H. Al-Khalifa, "Eye Movement Patterns as Indicators of Text Complexity in Arabic: A Comparative Analysis of Classical and Modern Standard Arabic," Journal of Eye Movement Research, vol. 18, no. 4, Art. no. 30, 2025. doi: 10.3390/jemr18040030.

[19] M.A. Ouassil et al., "Hybrid convolutional neural network-bidirectional long short-term memory model for Arabic sentence readability assessment," IAES International Journal of Artificial Intelligence, vol. 15, no. 3, pp. 2849–2862, 2026. doi: 10.11591/ijai.v15.i3.pp2849-2862.

[20] I. Baazeem, H. Al-Khalifa, and A. Al-Salman, "Integrating Linguistic and Eye Movements Features for Arabic Text Readability Assessment Using ML and DL Models," Computation, vol. 13, no. 11, Art. no. 258, 2025. doi: 10.3390/computation13110258.

[21] S. Berrichi, N. Nassiri, A. Mazroui, and A. Lakhouaja, "INTERPRETING THE RELEVANCE OF READABILITY PREDICTION FEATURES," Jordanian Journal of Computers and Information Technology, vol. 9, no. 1, pp. 36–52, 2023. doi: 10.5455/jjcit.71-1667559201.

[22] A. Basheer, S. Alawneh, T. Bostami, S. Daraghmeh, and A. Hawash, "Mapping Arabic Text Studying Material With The Most Suitable Student Grade," in 2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), 2022, pp. 259–267. doi: 10.1109/ICETSIS55481.2022.9888935.

[23] N. Nassiri, V. Cavalli-Sforza, and A. Lakhouaja, "MoSAR: Modern standard Arabic readability corpus for L1 learners," in ACM International Conference Proceeding Series, 2019. doi: 10.1145/3372938.3372961.

[24] T. Naous, M.J. Ryan, A. Lavrouk, M. Chandra, and W. Xu, "README++: Benchmarking Multilingual Language Models for Multi-Domain Readability Assessment," in EMNLP 2024 - 2024 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2024, pp. 12230–12266. doi: 10.18653/v1/2024.emnlp-main.682.

[25] S.S. Al-Thanyyan, and A.M. Azmi, "Simplification of Arabic text: A hybrid approach integrating machine translation and transformer-based lexical model," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 8, Art. no. 101662, 2023. doi: 10.1016/j.jksuci.2023.101662.

[26] S. Mohamed, "Text classification and gradation in Arabic textbooks," Language Learning Journal, vol. 52, no. 6, pp. 629–649, 2024. doi: 10.1080/09571736.2023.2213695.

[27] K. Kawar, E. Saiegh-Haddad, and S. Armon-Lotem, "Text complexity and variety factors in narrative retelling and narrative comprehension among Arabic-speaking preschool children," First Language, vol. 43, no. 4, pp. 355–379, 2023. doi: 10.1177/01427237221149800.

[28] H. Saddiki, K. Bouzoubaa, and V. Cavalli-Sforza, "Text readability for Arabic as a foreign language," in Proceedings of the 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), 2015, pp. 1–8. doi: 10.1109/AICCSA.2015.7507232.

[29] N. Khallaf, S. Sharoff, and R. Soliman, "Towards Arabic Sentence Simplification via Classification and Generative Approaches," in Proceedings of the Seventh Arabic Natural Language Processing Workshop, 2022, pp. 43–52. doi: 10.18653/v1/2022.wanlp-1.5.

[30] N. Al-Twairesh, A. Al-Dayel, H. Al-Khalifa, M. Al-Yahya, S. Alageel, N. Abanmy, et al., "MADAD: A Readability Annotation Tool for Arabic Text," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016.

[31] M. El-Haj, and P. Rayson, "OSMAN A Novel Arabic Readability Metric," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016.

[32] H. Saddiki, N. Habash, V. Cavalli-Sforza, and M. Al Khalil, "Feature Optimization for Predicting Readability of Arabic L1 and L2," in Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, 2018. doi: 10.18653/v1/W18-3703.

[33] M. Al Khalil, N. Habash, and Z. Jiang, "A Large-Scale Leveled Readability Lexicon for Standard Arabic," in Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020.

[34] Z. Jiang, N. Habash, and M. Al Khalil, "An Online Readability Leveled Arabic Thesaurus," in Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations, 2020. doi: 10.18653/v1/2020.coling-demos.11.

[35] N. Khallaf, and S. Sharoff, "Automatic Difficulty Classification of Arabic Sentences," in Proceedings of the Sixth Arabic Natural Language Processing Workshop, 2021.

[36] Y. SalahEldin, and C. Sabty, "Simplify: Automatic Arabic Sentence Simplification using Word Embeddings," in Proceedings of ArabicNLP 2023, 2023. doi: 10.18653/v1/2023.arabicnlp-1.35.

[37] M. El-Haj, S. Almujaiwel, D. Premasiri, T. Ranasinghe, and R. Mitkov, "DARES: Dataset for Arabic Readability Estimation of School Materials," in Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024, 2024.

[38] J. Liberato, B. Alhafni, M. Khalil, and N. Habash, "Strategies for Arabic Readability Modeling," in Proceedings of the Second Arabic Natural Language Processing Conference, 2024. doi: 10.18653/v1/2024.arabicnlp-1.5.

[39] B. Alhafni, R. Hazim, J. D. Pineros Liberato, M. Al Khalil, and N. Habash, "The SAMER Arabic Text Simplification Corpus," in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024.

[40] M. Basem, M. Younes, S. Ahmed, and A. Moustafa, "!MSA at BAREC Shared Task 2025: Ensembling Arabic Transformers for Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.42.

[41] M. Saeed, R. Waly, and A. Ashraf Hussein, "AMAR at BAREC Shared Task 2025: Arabic Meta-learner for Assessing Readability," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.45.

[42] K. N. Elmadani, B. Alhafni, H. Taha, and N. Habash, "BAREC Shared Task 2025 on Arabic Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.34.

[43] P. Elchafei, M. Osama, M. Rageh, and M. Abu-Elkheir, "GNNinjas at BAREC Shared Task 2025: Lexicon-Enriched Graph Modeling for Arabic Document Readability Prediction," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.36.

[44] N. Habash, H. Taha-Thomure, K. N. Elmadani, Z. Zeino, and A. Abushmaes, "Guidelines for Fine-grained Sentence-level Arabic Readability Annotation," in Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025), 2025. doi: 10.18653/v1/2025.law-1.30.

[45] S. Altammami, "Leveraging AI to Bridge Classical Arabic and Modern Standard Arabic for Text Simplification," in Proceedings of the New Horizons in Computational Linguistics for Religious Texts, 2025.

[46] A. Amel Nait Djoudi, P. Bellot, and A. G. Chifu, "LIS at BAREC Shared Task 2025: Multi-Scale Curriculum Learning for Arabic Sentence-Level Readability Assessment Using Pre-trained Language Models," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.51.

[47] F. Mohamed Emad Elden, "MorphoArabia at BAREC Shared Task 2025: A Hybrid Architecture with Morphological Analysis for Arabic Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.41.

[48] A. Abdou, "mucAI at BAREC Shared Task 2025: Towards Uncertainty Aware Arabic Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.44.

[49] M. Ayesh, "PalNLP at BAREC Shared Task 2025: Predicting Arabic Readability Using Ordinal Regression and K-Fold Ensemble Learning," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.47.

[50] A. Alhassan, A. Mohamed, and M. Elamin, "Phantoms at BAREC Shared Task 2025: Enhancing Arabic Readability Prediction with Hybrid BERT and Linguistic Features," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.49.

[51] B. Sapirstein, "Pixels at BAREC Shared Task 2025: Visual Arabic Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.48.

[52] S. Ahmad, "Qais at BAREC Shared Task 2025: A Fine-Grained Approach for Arabic Readability Classification Using a pre-trained model," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.43.

[53] S. Trigui, "STBW at BAREC Shared Task 2025: AraBERT-v2 with MSE-SoftQWK Loss for Sentence-Level Arabic Readability," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.50.

[54] A. M. Nazzal, "ZAI at BAREC Shared Task 2025: AraBERT CORAL for Fine Grained Arabic Readability," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.37.

[55] K. N. Elmadani, A. Mahmoud Wizani, H. Taha Thomure, and N. Habash, "A Large and Balanced Multi-Domain Arabic Corpus Annotated for Morphology, Syntax, and Readability," in Proceedings of the Fifteenth Language Resources and Evaluation Conference, 2026. doi: 10.63317/45f2o6t8piyi.

[56] N. Rabih, C. Qwaider, and T. Briscoe, "Can LLMs Control Readability? A Multi-Dimensional Evaluation Framework for CEFR-Controlled Arabic Generation," in Proceedings of the Joint Workshop on Readability and Text Simplification (READIxTSAR) @ LREC 2026, 2026. doi: 10.63317/48v7mxywgfja.

[57] H. Khalid, S. Shapsough, and I. Zualkernan, "Noise Steering for Controlled Text Generation: Improving Diversity and Reading-Level Fidelity in Arabic Educational Story Generation," in Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026), 2026. doi: 10.18653/v1/2026.bea-1.41.

[58] S. Alzubi, and R. Reynolds, "Transformer-based readability classifiers are worse than you think: Evidence from cross-domain Arabic readability assessment," in Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026), 2026. doi: 10.18653/v1/2026.bea-1.52.

[59] A. Bahloul, "Syntaxa at BAREC Shared Task 2025: BERTnParse - Fusion of BERT and Dependency Graphs for Readability Prediction," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.35.

[60] Y. Bestgen, "SATLab at BAREC Shared Task 2025: Optimizing a Language-Independent System for Fine-Grained Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.40.

[61] S. Ibrahim, M. Rafiul Biswas, M. Bessghaier, and W. Zaghouani, "MarsadLab at BAREC Shared Task 2025: Strict-Track Readability Prediction with Specialized AraBERT Models on BAREC," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.39.

[62] N. Rabih, "Noor at BAREC Shared Task 2025: A Hybrid Transformer-Feature Architecture for Sentence-level Readability Assessment," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.46.

[63] S. Sibaee, O. Nacar, Y. Alhabashi, A. Ammar, and W. Boulila, "ANLPers at BAREC Shared Task 2025: Readability of Embeddings Training Neural Readability Classifiers on the BAREC Corpus," in Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks, 2025. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.38.

[64] M. S. Alassaf, H. H. Hamadallah, A. Almuzaini, A. M. Aloufi, K. N. Al-Turki, A. S. Khoshhal, M. A. Alsulaimani, and R. Eshky, "Assessment of the Arabic patient-centered online information about orthodontic pain: A quality and readability assessment," PLOS ONE, vol. 19, no. 5, Art. no. e0303308, 2024. doi: 10.1371/journal.pone.0303308.

APPENDIX A: REPRODUCIBLE SEARCH COVERAGE

The full database-specific strings, filters, dates, and counts are supplied in the Search Log sheet of the supplementary workbook. The ACL update replicated the original Arabic and concept blocks in official title-and-abstract metadata, then ran a distinct sensitivity search using standalone "readability." Arabic-script strings are provided as plain UTF-8 cells to avoid bidirectional corruption in typeset syntax. The BAREC cluster check and ACM DOI-prefix verification are documented separately.

TABLE V. REPRESENTATIVE STUDY-LEVEL EVIDENCE (FULL 61-STUDY TABLE IN SUPPLEMENT)

Study/family	Population, data, or design	Principal finding	Main limitation
ATSRA-007	54 university students; four textbook passages; Cloze	All passages at frustration level; passage means 0.46–0.66	One institution/textbook
ATSRA-077	30 preschool children; four MSA/Palestinian Arabic narratives	Variety and complexity affected comprehension/retelling; MSA higher in tested design	Small advantaged sample; four stories
AraEyebility family	Later corpus/analyses share data; CA–MSA comparison n = 15	Gaze features and register contrasts reveal processing differences	Overlap with the earlier 41-reader pilot is unverified
BAREC family	69,441 sentences; 1M+ words; 19 levels; 18 shared-task/overview papers	Best QWK 0.875 sentence/0.874 document; annotation QWK 0.818	Common benchmark; no learner outcome
SAMER family	Approximately 26,000 lemmas; 159,000 original-corpus words	Supports word/fragment models and two target simplifications	Linked resources; literary-domain limits
DARES family	13,335 school-text instances; grades 1–12	Coarse weighted/macro F1 0.91/0.79; fine 0.68/0.55	Saudi curriculum; later version overlaps
ATSRA-004	ATSC, 500 article/simplification pairs	SARI≈0.73; BLEU≈0.65; METEOR≈0.68	Automatic metrics; no learner test
ACL-007	4,258 word records; 8/69/765 sentence-pair sets; expert ratings	Identification F1 0.88; simplicity–meaning–grammar trade-off	Experts, not intended learners
ACL-029	JSTC, 82,512 segments from 240 textbooks; DARES/BAREC external test	CAMeLBERT QWK 0.830 in-domain, 0.085 cross-domain; XGBoost cross-domain 0.240	Shows severe domain dependence
L2/GLOSS family	Repeated learner-oriented corpora and feature studies	Main L1/L2 accuracy 94.8%/72.4%; augmented L2 kNN 74.1%	Different feature/model settings; no new comprehension sample

An Adaptive AI-Driven Code Optimization Framework Using Machine Learning and Explainable AI for Scalable Software Systems

Kavya B K[1]*, S.Annamalai[2], A Suresh Kumar[3]
Research Scholar, Department of Computer Science and Engineering,
Jain (Deemed-to-be University), Bangalore, Karnataka, India[1]
Associate Professor, Department of Computer Science and Engineering,
Jain (Deemed-to-be University), Bangalore, Karnataka, India[2]
Professor, Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, Karnataka, India[3]

***Abstract*—The increasing complexity of embedded computing systems has made it difficult to achieve efficient code execution while effectively utilizing limited hardware resources. Conventional optimization methods mainly rely on compiler-driven techniques and fixed transformation rules, which may not adapt well to different embedded architectures and varying workload conditions. This research introduces a hardware-aware AI-driven code optimization framework that integrates Machine Learning (ML) with Explainable Artificial Intelligence (XAI) to improve the efficiency and scalability of embedded software systems. The framework analyzes software characteristics together with hardware-related parameters, including processor usage, memory requirements, computational workload, and resource constraints, to identify performance limitations and select suitable optimization strategies. A machine learning model is employed to predict appropriate code transformation techniques based on application behaviour and underlying hardware conditions. The XAI module further analyzes the optimization process by identifying the contribution of individual features, providing clearer explanations for the selected optimization decisions. Experimental results show that the proposed framework reduces execution time by up to 30%, decreases memory consumption by 25%, and lowers CPU utilization by approximately 15% while maintaining performance as code complexity increases. These findings demonstrate the capability of the framework to support efficient and transparent optimization of embedded software under resource-constrained computing conditions.**

Keywords*—Hardware-aware optimization; artificial intelligence; code optimization; machine learning; explainable AI; scalability

I. Introduction

The increased size of large-scale software systems and high-performance applications has posed a serious problem of how to maximize the efficiency of code, according to the continually growing requirements of the new computer world. With the advancement of software systems, the demand for an effective code optimization tool that can not only enhance the level of computational capabilities but also optimise the use of resources has never been more imperative [1]. The traditional code optimization techniques, which are mostly static in nature discipline specialise in a single area of performance with regard to code execution time or memory [2]. Although these methods have been found effective in better-controlled environments, they are at times inadequate in dealing with the dynamic nature of modern software systems, which need to perform effectively in a variety of platforms, workloads and usage scenarios [3]. Limitations of the use of traditional and optimisation methods are also evident in solving the disparate and dynamic demands of contemporary apps [4]. To take some examples, in traditional systems that do not support environmental variability (hardware differences) or dynamic runtime variability, the performance may not be optimal when applied into practice [5]. Moreover, most of the current methods cannot keep up with the size and complexity of the contemporary distributed systems or cloud-based systems where the amount of data and the multitude of tasks are likely to create new bottlenecks that cannot be scaled or forecasted by the existing models [6]. These constraints demonstrate the necessity to approach the optimization of codes more dynamically and flexibly [7]. To address these difficulties, it is already stated that there is growing interest in adaptive, scalable, and interpretable optimization models that could offer more beneficial and resilient answers. The main benefit of these types of models is that they are able to keep learning based on the runtime data and adjust optimization strategies based on real-time evidence [8]. Additionally, scalability will be crucial given the changes the software systems will experience in order to accommodate bigger datasets, increased concurrent usage, and complexity of the codebase [9]. Hence, it is of utmost importance to create a framework that codes can be optimized as well as can scale as with the development of the system, and can adjust to evolving needs. The combination of Artificial Intelligence (AI) and Explainable AI (XAI) in contemporary software engineering is a prospective approach to tackling the limitations of conventional optimization methods [10]. Code analysis can be made automatic, and machine learning algorithms can spot areas of performance bottlenecks where improvements can be made and suggest optimization strategies in recommendations based on historical data, using AI [11]. In addition, XAI will introduce a degree of transparency to the optimization procedure, as it can give the reasons why the AI models made a decision that can be read and understood by humans [12]. This interpretability is paramount since it enables developers to know the reasons behind some optimization strategies being taken, and this leads to the recommendations

*Corresponding author.

being easier to believe in, so that it becomes easier to implement [13]. This research aims to establish a scalable and efficient code optimization framework that targets enhancing the effectiveness of the computations and the use of resources. This framework utilizes the strength of AI to proceed to dynamically optimise code within a range of software systems, as well as being scalable and interpretable by XAI. The proposed framework contributes to intelligent software engineering by enabling automated and hardware-aware code optimization. It supports the development of efficient embedded systems with reduced computational resource requirements, benefiting applications such as IoT, healthcare devices, autonomous systems, and smart technologies. The research also promotes sustainable computing by improving system efficiency and resource utilization.

The remainder of this manuscript is organized as follows: Section II presents the literature review, Section III describes the proposed framework architecture, and Section IV introduces the mathematical model for optimization. Section V discusses the algorithms, and Section VI presents the implementation details, while Section VII presents the experimental evaluation. Section VIII discusses the results, and Section IX concludes the study with future research directions.

II. Literature Review

A. Traditional Code Optimization Techniques

Conventional optimization techniques that are used in the optimization of codes focus mainly on compiler-based techniques, which entail loop unrolling, dead code elimination, and inlining [14]. These methods are directed to enhance performance through alteration of the code during the compilation stage [15]. Loop unrolling is used to minimize the overhead of loop control, and dead code elimination is used to eradicate code that never gets executed to produce output, hence unnecessary computations. Inlining is the replacement of function calls by the body of the functions in order to get rid of the performance cost of calling functions [16]. Such optimizations are adapted to the relatively simple cases of applications whose complexity in code and environmental effects are relatively constant. The shortcomings of the traditional static optimization techniques are, however, immediately evident in cases of more dynamic and large-scale systems. The solution of static optimization does not provide the flexibility in response to the change inustome conditions, hardware, or different loads [17]. As an example, performance increases obtained by loop unrolling might not be realized with the same effectiveness in systems with a mixed hardware configuration, or they cannot be scaled effectively using multi-threaded systems [18]. As a result of diversified and unpredictable workloads, dynamic optimization tools that adapt the system at runtime to real-time data and changing operating conditions have become more crucial for modern computing applications. Conventional optimization techniques mainly rely on compiler-based transformations and prior optimization rules that typically offer restricted flexibility across various software architectures and runtime scenarios. While machine learning-based optimization methods have improved automated decision-making, it is still challenging to generalize these models even with large datasets, and furthermore difficult to build interpretable models of complex software systems that can scale easily. A comparison of existing approaches without AI-based code optimization, along with their drawbacks, is shown in Table I.

TABLE I. Comparative Analysis of Existing AI-Based Code Optimization Approaches

Study Approach	Methodology	Key Contributions	Limitations
Traditional Compiler Optimization	Rule-based transformations, static analysis, heuristic optimization	Provides reliable optimization techniques such as loop transformation, dead code elimination, and instruction scheduling	Limited adaptability for complex software systems; optimization decisions are predefined and lack learning capability
Reinforcement Learning-Based Code Optimization	RL models for compiler parameter tuning and optimization sequence selection	Enables automated exploration of optimization strategies and improves optimization decisions through feedback	Requires extensive training data and computational resources; optimization decisions are difficult to interpret
Machine Learning-Based Performance Prediction	Supervised learning models using software metrics and execution features	Predicts performance bottlenecks and identifies optimization opportunities using learned patterns	Performance depends strongly on feature quality and training dataset diversity
Deep Learning-Based Program Optimization	Neural networks for code representation learning and optimization prediction	Captures complex relationships among program structures and execution behavior	High computational complexity and limited explainability reduce practical adoption
AI-Assisted Code Transformation	Automated generation and modification of optimized code segments	Reduces developer effort by suggesting optimization actions	Generated transformations may not always maintain optimal performance across different applications
Explainable AI-Based Software Analysis	XAI methods such as feature importance analysis and model interpretation	Improves transparency of AI decisions and supports developer trust	Explainability mechanisms introduce additional computational overhead

1) AI and machine learning in code optimization: The development of Artificial Intelligence (AI) and Machine Learning (ML) has provided new opportunities in code optimization. One such use is in ML-based performance prediction, a machine learning model trained to estimate the time required to execute it, the amount of memory required to execute it, or other performance measurements [19]. Recent studies have explored reinforcement learning-based approaches for code optimization by learning optimization decisions through interaction with the execution environment and feedback signals. However, several other AI-based techniques, including supervised machine learning, deep learning, and evolutionary optimization methods, have also been investigated for improving software performance under different

optimization objectives [20]. This will allow adaptive optimization, especially in dynamic systems where the performance requirements change.

2) Scalability and resource-aware optimization: Scalability and optimization of resources based on resources have become important issues as software systems increase in size and complexity. The distributed and cloud systems performance tuning needs dynamic approaches which will be able to adapt to environmental variability, most of which is usually unpredictable [21]. Local machine optimizations might fail in a distributed system with much latency and bandwidth variability, as well as resource allocation. Such an environment as clouds is characterized by the varying availability of resources and therefore the need to optimize its resources over time and with each change. Additionally, optimization soon with a focus on energy efficiency and memory consciousness. The diverse landscape of computing platforms, spanning from resource-constrained mobile and edge devices to high-performance computing clusters, has made clear the need for optimization techniques that can trade off energy consumption and memory utilization while preserving computational performance. The energy-efficient optimization approaches focus on improving resource utilization with acceptable execution quality, in particular for embedded and edge computing environments where hardware limitations are high. While prior work showed that AI-based optimization methods enable software performance improvements, several challenges have yet to be overcome. The existing machine learning-based approaches predominantly address optimization accuracy or execution improvement, but do not effectively account for the aspects of resource constraints, adaptability, and scalability. Additionally, a majority of AI-based optimization frameworks are black-box models that do not explain the rationale behind chosen optimization actions to developers. The absence of visible mechanisms for decision-making reduces users' trust and accounts for possible adoption failures in software systems that are required to be reliable. Table II presents the identified research gaps in existing approaches and highlights the contributions and novelty of the proposed adaptive AI-driven framework.

TABLE II. RESEARCH GAP ANALYSIS AND PROPOSED FRAMEWORK CONTRIBUTION

Research Aspect	Existing Approaches	Identified Research Gap	Proposed Framework Contribution
Optimization Adaptability	Most approaches depend on fixed rules or predefined optimization strategies	Limited capability to adapt optimization decisions according to changing software characteristics	Develops an adaptive AI-driven optimization mechanism using ML-based decision prediction
Performance Bottleneck Identification	Existing methods mainly rely on manual profiling or static analysis	Inefficient identification of complex performance issues in large software systems	Uses machine learning models to predict optimization opportunities from code characteristics
Resource Optimization	Many studies focus mainly on execution speed improvement	Limited consideration of memory consumption and CPU utilization during optimization	Incorporates resource-aware optimization objectives for balanced performance improvement
Scalability	Existing optimization methods often evaluate small benchmark programs	Limited validation for large and complex software systems	Introduces scalability analysis to evaluate performance consistency with increasing code complexity
Optimization Transparency	Many AI-based approaches provide black-box optimization decisions	Lack of explanation reduces developer confidence and adoption	Integrates Explainable AI to provide interpretable optimization decisions
Generalization Capability	Models are often trained for specific programming environments or datasets	Difficulty applying optimization models across diverse software systems	Provides an adaptive framework designed for broader software environments

3) Research gap: Although existing studies have confirmed the potential of AI techniques for software optimization, current approaches still lack a comprehensive framework that simultaneously addresses adaptability, resource awareness, scalability, and interpretability. Most of the existing optimization methods tend to either improve upon a very specific performance metric or provide automated recommendations on what ~should-be-done to optimize without thorough information on the drivers behind the selected optimization choices. Additionally, the use of optimization frameworks that dynamically adapt to different characteristics in each software while preserving transparency and interpretability has been less explored. This research proposes a new adaptive AI-powered code optimization framework incorporating Machine Learning and Explainable AI for large-scale software systems to overcome these limitations. The proposed framework is distinct from existing work in that it synergistically integrates performance bottleneck identification, intelligent optimization strategy selection, resource-aware evaluation, scalability analysis, and explainable decision support within a single architecture. The integrated approach not only allows a focus on efficient optimization but also increases the transparency, reliability, and practicality of AI-assisted code optimization in complex computing environments.

III. PROPOSED FRAMEWORK ARCHITECTURE

1) Overall framework design: The proposed framework is set to automate and optimize the code optimization process via the incorporation of an integrated workflow of steps.

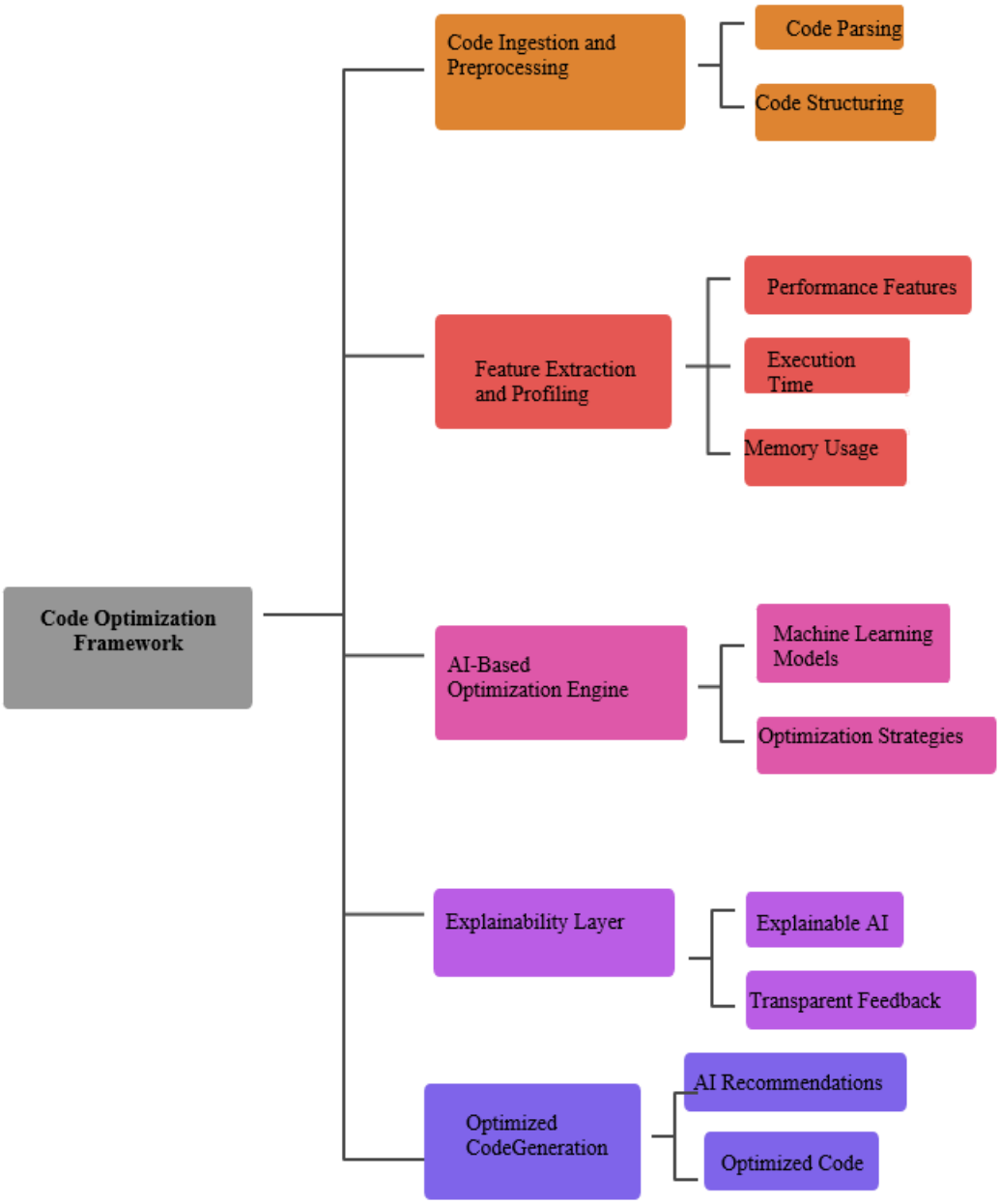


Fig. 1. Proposed system framework design.

The code ingestion and preprocessing stage represents the initial phase of the proposed framework, where source code is analyzed and transformed into a structured representation through parsing and normalization techniques. This transforms input heterogeneous code into a uniform code format to facilitate feature extraction and performance assessment. In the feature extraction and profiling stage, following the preprocessing, this step determines key performance indicators derived from the structure of our program as well as its execution behaviour. The features extracted are characteristics of control flows, data dependencies, patterns for memory utilization, and properties regarding execution behaviour or performance parameters related to hardware. Combined, these properties provide a holistic description of the software and hardware characteristics used by the optimization model to locate performance bottlenecks and choose appropriate optimization methods. Profiling tools are also used to understand the performance of the code under various conditions, and this allows for identifying the possible bottlenecks and inefficiencies. The high-level architecture shown in Fig. 1 shows the proposed code optimization framework. The framework revolves around the AI-based optimization engine. This engine uses machine learning models to forecast performance problems and suggestions. Using the collected past performance data and performance metrics, the engine is able to propose optimizations like changes in the algorithm, memory utilization, or even parallelization patterns. The machine learning models that would be applied in this step are altered to respond to various kinds of coding and optimization objectives to make the engine work in different cases. An explainability layer is added into the

framework in order to make the optimization decisions transparent and understandable. This layer exploits Explainable AI (XAI) methods to give human-readable explanations of the optimization decisions taken by the AI-based engine. The explainability layer will enable the developers to understand the reasoning behind certain optimizations that have been chosen, which will build more confidence in the system.

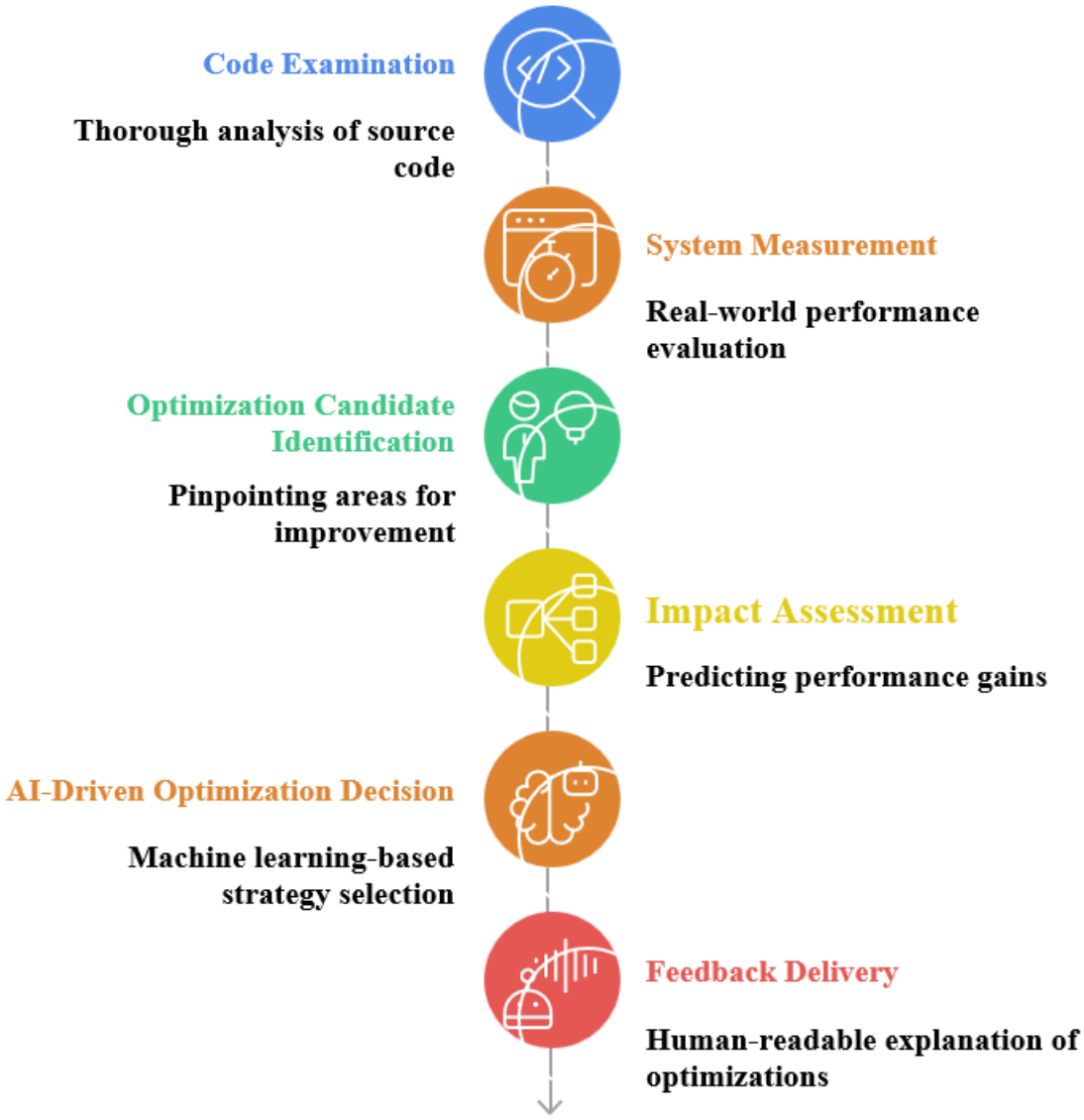


Fig. 2. Framework workflow.

Finally, the last element of the framework is to generate optimized code. Once the optimization decisions are made, the framework uses the possible transformations suggested to the source code to generate a version of the source code that is optimized. This code that has been generated is subjected to strict validation so that functional consistency remains intact and the performance-related improvements are achieved.

2) Framework workflow: The process of the proposed framework is started by the source code analysis step, where the system analyzes the raw code and identifies the key features and calculates the performance features. This step points out areas where there might be a problem, e.g., inefficient loops, unnecessary calculations, or excessive use of memory. It is then followed by performance profiling, where the time spent by the code to run, its memory utilization, and all other pertinent metrics are measured. The performance measures of the code are obtained, and the profiling tools across various execution environments give a complete picture of the behaviour of the code in different conditions. During the candidate identification phase of optimization, the framework establishes certain areas within the code that could be optimized. This step entails comparing the profiling data with established patterns of inefficient code, where improvements that could be made on the optimization of performance could be identified. Upon identifying the candidates, mathematical assessment is conducted in order to determine the effect of different optimization approaches. There are some mathematical models that assess the trade-offs among all the various options, which

can be in terms of execution time, memory consumption, and resource usage. This assists in choosing the most effective method of optimization. The workflow of the proposed code optimization framework is presented in Fig. 2 step by step. The framework will then move on to an AI-based optimization decision where machine learning algorithms will choose the most appropriate optimizations, depending on the candidates identified and the outcomes of the mathematical analysis. The process of decision-making is dynamic as it changes in response to the situation and the preference of the codebase. Lastly, there is the explainable feedback generation stage, which offers a human-readable explanation of the decisions made during optimization by AI. This feedback serves to notify the developers about the rationale behind the proposed changes and to make sure that the optimizations are directed towards acceptability of the desired performance goals.

IV. Mathematical Model for Code Optimization

The code optimization mathematical model is designed to be able to solve the problem of how to increase the efficiency of computer activities and reduce the consumption of resources, such as the time of execution, the amount of memory, and energy. The optimization process is defined in terms of a set $C = \{c1, c2, \dots, cn\}$, where each c_i represents a code segment that can be analyzed and optimized. For each code segment, three key performance indicators are considered: $\boldsymbol{T}(c_i)$, which denotes the execution time of the segment; $M(c_i)$, which represents the memory consumption; and $\boldsymbol{E}(c_i)$, the energy or resource cost associated with the segment. These parameters are crucial in evaluating the performance and resource requirements of each code segment.

The goal of optimization is to lower the cost of computation whilst keeping the computation functional. The following equation gives this objective:

$$min \sum_{i=1}^{n} (\alpha T(c_i) + \beta M(c_i) + \gamma E(c_i) \quad (1)$$

where, α, β, γ are weight factors that control the importance of execution time, memory, and energy in the optimization process. Such weights can be used to prioritize some of the performance metrics, according to the needs of the system. This is aimed at relaxing the total resource usage so that the code being optimized will be more efficient in these crucial parameters. To solve the optimization problem, a few restrictions are required to make sure that the application of code segments will not violate the system requirements. The functional equivalence constraint states that the optimal code will act in the same way as the original code, and this is mathematically defined as ending up as:

$$f(c_i) = f(c_i') \quad (2)$$

where, $f(c_i)$ and $f(c_i')$ represent the functionality of the original and optimized code segments, respectively. Such a limitation ensures that the software does not change the desired functionality during optimization. In addition, the resource constraint can guarantee that the critical usage of the optimized code is less than a predetermined maximum amount:

$$M(c_i') \leq M_{max} \quad (3)$$

where, $M(c_i')$ is the memory consumption of the optimized code, and M_{max} is the maximum allowable memory usage. Finally, there is the scalability constraint, which makes sure that the optimized code can scale accordingly without major performance loss. This is expressed as:

$$T(c_i') \leq T(c_i)\ X\ \delta \quad (4)$$

where, $T(c_i')$ represents the execution time of the optimized code, $T(c_i)$ represents the execution time of the original code, and δ is a scalability factor that defines the acceptable level of performance loss during optimization. An AI-based model is used to get the optimization choices of each code part considered taking into account the performance measurements and scalability characteristics. The decision formulated is the optimization decision as follows:

$$O(c_i) = \mathrm{AI(T, M, E, S)} \quad (5)$$

where, $O(c_i)$ represents the optimization decision for the code segment, and $T, M,$ and E are the performance metrics (execution time, memory, and energy). S symbolizes scalability properties, i.e. code size, and the multiple-core property, and such properties affect the optimization strategy. The proposed adaptive AI-driven code optimization framework is formulated as a multi-objective optimization problem that minimizes execution overhead while maximizing resource efficiency. The software program is represented as:

$$P = \{X, F, R\}$$

where, X represents extracted code features, F represents functional characteristics, and R represents resource utilization parameters.

The optimization process transforms the original program P into an optimized version P^*:

$$P^* = O(P, S^*) \quad (6)$$

where, O represents the optimization operation and S^* represents the optimal optimization strategy.

- Multi-Objective Optimization Function

The optimization objective is defined as:

$$\min J\ (P) = w_1 T + w_2 M + w_3 C \quad (7)$$

where:

$$w_1 + w_2 + w_3 = 1 \quad (8)$$

$$T = \frac{T_{exec}}{T_{max}} \quad (9)$$

$$M = \frac{M_{use}}{M_{max}} \quad (10)$$

$$C = \frac{CPU_{use}}{CPU_{max}} \quad (11)$$

where $T, M, and\ C$ represent normalized execution time, memory consumption, and CPU utilization, respectively.

The optimal solution is obtained by:

$$P^* = arg \min_{p_i \in \Omega} J(Pi) \quad (12)$$

where, Ω represents the optimization solution space.

- Machine Learning-Based Optimization Prediction

The extracted software feature vector is represented as:

$$X = [x_1, x_2, x_3, \dots \dots, x_n] \quad (13)$$

where features include:

$$X = \{CC, IC, LF, DM, MA\} \quad (14)$$

where:

CC = cyclomatic complexity

IC = instruction count

LF = loop frequency

DM = dependency metrics

MA = memory access characteristics

The ML model predicts optimization probability:

$$Y = f_\theta(X) \quad (15)$$

where, Y represents the predicted optimization class and f_θ represents the trained AI model.

The optimal optimization strategy is selected as:

$$S^* = \max_{Si} P(Si|X) \quad (16)$$

where:

$$P(Si|X) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (17)$$

- Performance Improvement Model

Execution time improvement:

$$\Delta T\ (\%) = \frac{T_b - T_b}{T_b}\ X\ 100 \quad (18)$$

Memory reduction:

$$\Delta M\ (\%) = \frac{M_b - M_a}{M_b}\ X\ 100 \quad (19)$$

CPU utilization reduction:

$$\Delta CPU\ (\%) = \frac{CPU_b - CPU_a}{CPU_b}\ X\ 100 \quad (20)$$

where, subscripts b and a represent the values before and after optimization.

Overall efficiency improvement:

$$E = \alpha\Delta T + \beta\Delta M + \gamma\Delta CPU \quad (21)$$

subject to:

$$\alpha + \beta + \gamma = 1$$

- Resource-Aware Optimization Model

The resource utilization vector is defined as:

$$R = [r_{cpu}, r_{mem}, r_{io}] \quad (22)$$

The resource efficiency function is:

$$RE = \frac{R_b - R_a}{R_b} \quad (23)$$

The optimization constraint is:

$$R_a \leq R_{limit} \quad (24)$$

- Scalability Model

The scalability factor is defined as:

$$S_c = \frac{P_{opt}(n_2)}{P_{opt}(n_1)} \quad (25)$$

where, n_1 and n_2 represent different software sizes.

Optimization stability:

$$S_t = 1 - \frac{|E_{n2} - E_{n1}|}{E_{n1}} \quad (26)$$

where, E represents optimization efficiency.

- Explainable AI Model

The contribution of each feature toward the optimization decision is represented as:

$$I_i = \frac{|\varphi_i|}{\sum_{j=1}^{n} |\varphi_j|} \quad (27)$$

where, φ_j represents the feature contribution value.

The final AI optimization decision is expressed as:

$$D = f_\theta(X) + XAI(X) \quad (28)$$

This model makes sure that the optimization decisions are dynamic, flexible, and able to enhance the code performance without undermining the system resources integrity and functional correctness.

A. Algorithms

The following algorithms (Algorithm 1-4) define the most important stages in the optimisation procedure of a code, using machine learning and explainable AI to optimise its performance without jeopardising its functional qualities.

Algorithm 1: Code Feature Extraction

Input: Source Code Output: Feature Vector

Steps:

Parse the Abstract Syntax Tree (AST): Convert the source code into an AST, which shows the structure of the code.

Extract Control Flow and Data Flow Features: Identify how control moves through the code (which parts are executed and in what order) and how data is passed between variables and functions.

Compute Performance Metrics: Measure key performance factors like:

$T(c_i)$ (Execution time of the code segment)

$M(c_i)$ (Memory usage of the code segment)

$E(c_i)$ (Energy/resource cost of the code segment)

Generate Feature Vector: Combine the extracted features and performance metrics into a structured feature vector:

$$\text{Feature Vector} = [T(c_1), M(c_1), E(c_1), T(c_2), M(c_2), E(c_2), \dots\dots.] \quad (29)$$

Algorithm 2: AI-Based Optimization Selection

Input: Feature Vector Output: Optimization Strategy

Steps:

Load Trained Machine Learning Model: Load the pre-trained model that can predict which parts of the code need optimization.

Predict Performance Bottlenecks: Use the feature vector to predict which parts of the code are inefficient.

$$\mathbf{Bottlenecki} = \mathbf{Predict}(\mathbf{T(ci)}, \mathbf{M(ci)}, \mathbf{E(ci)})$$

Select Optimal Transformation: Choose the best optimization strategy for each identified bottleneck, such as reducing memory usage or improving execution time.

Rank Optimization Candidates: Rank the possible optimizations by their potential to improve the code's performance.

$$\text{Rank}(01, 02, 03, \dots) \quad (30)$$

Algorithm 3: Explainable Optimization Mapping

Input: Optimization Decision

Output: Human-readable Explanation

Steps:

Identify Influential Features: Identify which features (like execution time, memory usage) most affected the optimization decision.

$$\mathbf{Influential\ Features} = \{\mathbf{T(ci)}, \mathbf{M(ci)}, \mathbf{E(ci)}\} \quad (31)$$

Apply XAI Technique (e.g., SHAP/LIME): Use an Explainable AI (XAI) technique to create an explanation for the optimization.

$$\mathbf{Explanation} = \mathbf{XAI}(\mathbf{Influential\ Features}) \quad (32)$$

Generate Explanation: Create a readable explanation of why a particular optimization was chosen.

Map Explanation to Code Lines: Link the explanation back to specific parts of the code to show where the optimizations are applied.

Algorithm 4: Optimized Code Generation

Input: Selected Optimization Strategy

Output: Optimized Code

Steps:

Apply Transformation Rules: Apply the selected optimization strategy to the code.

$$\text{Optimized Code} = \text{Apply Transformation}(0) \quad (33)$$

Validate Correctness: Ensure that the optimized code still behaves the same as the original.

$$f(c_i) = f(c_i') \quad (34) \text{ (Functional equivalence)}$$

Reprofile Optimized Code: Measure the performance of the optimized code.

$$T(c_i')M(c_i')E(c_i') \quad (35)$$

(New performance metrics)

Output Optimized Code: Output the final optimized version of the code, ready for execution.

V. Implementation Details

The process of implementing the proposed code optimization framework implies several important technologies and tools. The framework has been developed mainly in Python, as it has a great number of support for machine learning and data analysis libraries. Also, components of the structure that demand high-performance implementation, such as memory-consuming activities or low-level system interactions, are instead done in C / C++ so as to have maximum execution speed as well as resource control. Java is also applied when working with larger programs or adhering to enterprise-level programs, in which scalability and cross-platform compatibility are needed. The machine learning models utilized in the optimization process are Random Forests because of their capability to process large volumes of data and give strong predictions on performance. Pattern recognition of complex patterns in the code structure is done by the use of Neural Networks, and thus, the framework is also able to pick inefficiencies that the traditional models might not be able to pick. Also, Reinforcement Learning (RL) is incorporated to respond dynamically to real-time feedback to optimize the set of strategies to better fit its strategy as it learns over time through optimization. To profile the performance of the code, there are a number of profiling tools that are employed to collect the execution time, memory usage, and energy consumption information. Profiling tools of Embedded C/C++ code, gprof and Valgrind, are used, whereas Python code profiling is conducted by cProfile and line_profiler. The tools are used to identify bottlenecks in the performance and those parts of the code that consume a lot of resources and that need to be optimized. The machine learning models are trained based on a diverse dataset of an equally varied number of codebases, which provides a robust backbone for predicting performance issues across various programming languages and different architectures. To maintain transparency when optimizing the process, XAI libraries are introduced into the framework, such as SHAP and LIME. Such libraries give explainable results on the optimization choices, giving the developer explicit and comprehensible feedback about the rationale behind the removal of particular code optimizations, increasing the trust and usefulness of the framework to developers.

VI. Experimental Evaluation

The effectiveness of the suggested code optimization framework is tested experimentally on the grounds of some major performance criteria. The reduction in execution time is the percentage of time that the optimized code executed faster than the original. This measure is essential to evaluate how fast the program can be implemented by the framework. The reduction in memory usage is measured by comparing the memory consumption prior to optimization and after it, and it is necessary to have minimum resource usage. CPU usage is also observed to make sure that the optimization process does not cause an increase in the level of computational overhead. The scalability of the proposed framework is evaluated by applying it to software systems with varying code sizes and complexity levels. This assessment examines the ability of the framework to maintain optimization effectiveness as the size of the codebase increases and as applications become more computationally demanding. The experimental evaluation is conducted using established benchmark datasets containing diverse applications, ranging from small-scale programs with limited computational requirements to large-scale systems with complex functionalities. This diversity enables a comprehensive

assessment of the optimization capability under different workload conditions. The experiments are performed on a multi-core computing platform with sufficient memory resources using a Linux-based environment. The implementation supports widely used programming languages, including Python and Embedded C/C++, to evaluate the adaptability of the framework across different software development scenarios. Implementing the profiling and optimization process is done with the help of tools such as gprof, Valgrind, and cProfile, depending on the programming language one is doing. In order to determine the effectiveness of the proposed framework, the comparative analysis of the traditional compiler optimization approaches and the proposed AI-driven framework is conducted. Existing optimizations (e.g., loop unrolling, inlining, dead code elimination, etc.) are contrasted with the dynamic optimizations based on machine learning provided by the proposed framework. The findings show that the proposed method can produce substantial speed and memory space savings on activities and improve scalability, as this cannot be done using conventional approaches.

VII. Results and Discussion

The code optimization framework, in terms of its experimental assessment, has established substantial performance gains in relation to a variety of indicators, such as the reduction of performance in terms of time performance, memory consumption, and CPU usage. The execution time reduction of different applications is presented in Fig. 3, wherein it is observed that there is a noticeable change in execution time optimization in all applications. The AI-based structure is better than the traditional optimization techniques, and the time reduction is up to 15-30. The strategies have been optimized, namely through memory management and algorithmic optimization, which helps in faster execution, as demonstrated by performance-intensive apps like "Image Processing" and "Machine Learning" among others. On the same note, Fig. 4 indicates the amount of memory usage saved with the proposed framework. The findings suggest that memory usage is reduced significantly by 10-25% percentages amongst applications. Applications that are known to consume high levels of memory, such as in the case of Database Query and Web Server, are highly affected by optimizations such as data structure tuning and memory pooling. These optimizations allow a more efficient utilization of system resources, which eventually limits the memory pressure and avoids memory bottlenecks. Fig. 5 shows the CPU utilization of the pre-optimization and the post-optimization. The smaller usage of CPU can be significantly mentioned, and the AI-based optimization methods can reduce CPU consumption by 15%. This is especially advantageous for long-running applications and applications with heavy load as it reduces the total amount of energy consumed by the system. Minimum use of CPU is also an indication that the framework is one that does not only optimizes performance but also minimises the cost of running the software application.

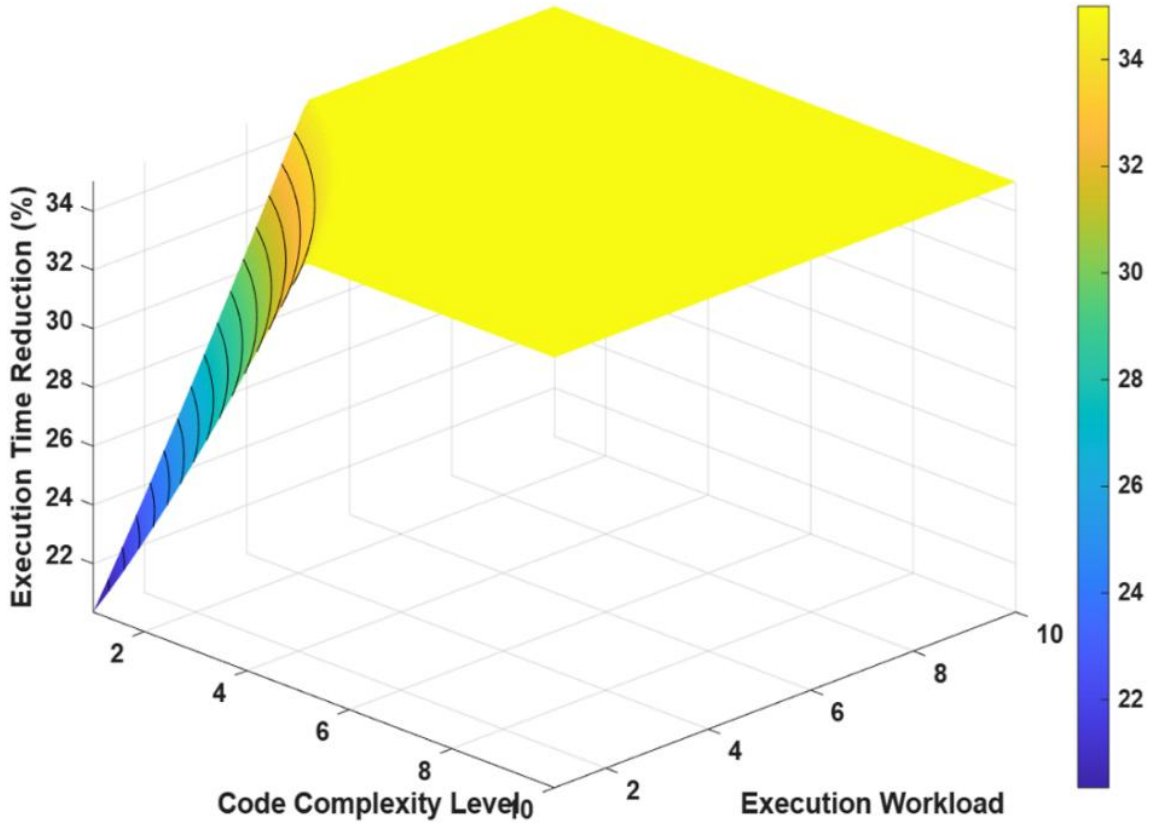


Fig. 3. Execution time reduction (%).

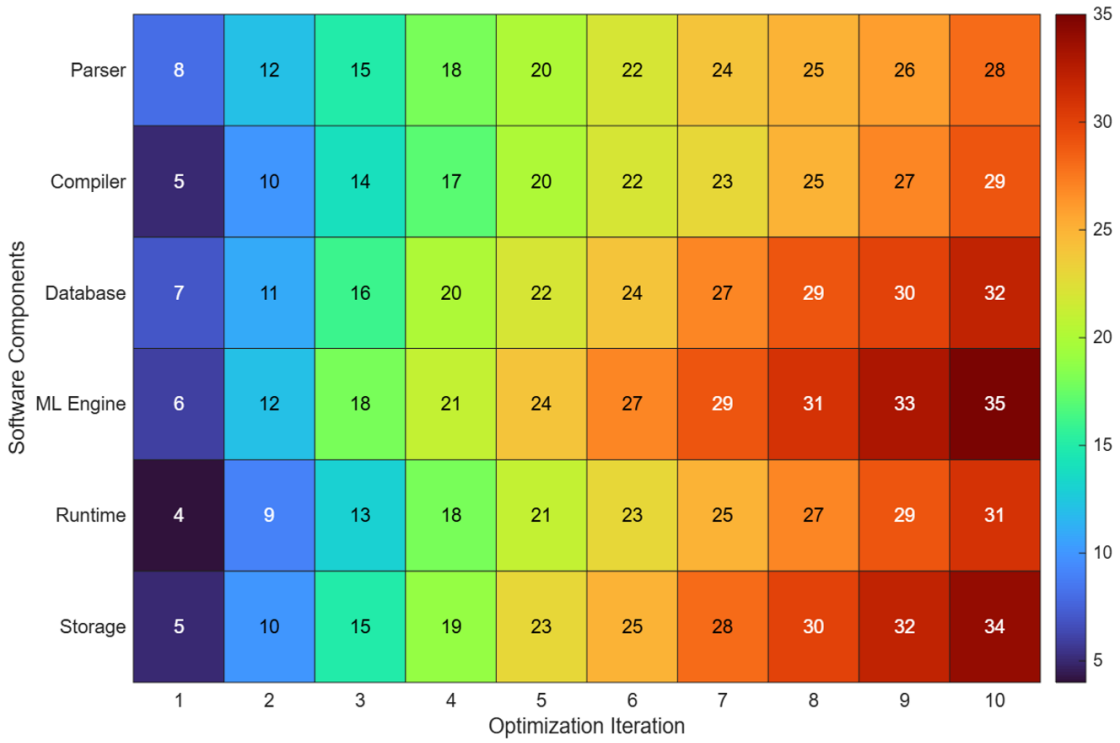


Fig. 4. Memory optimization heatmap.

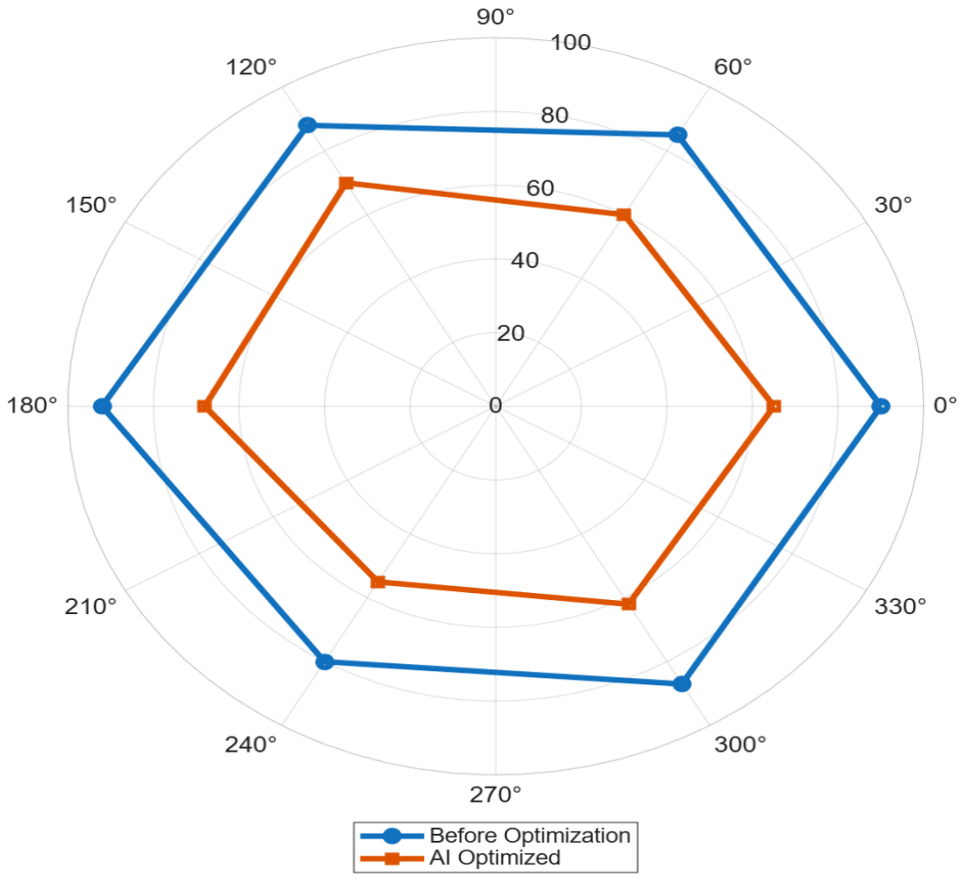


Fig. 5. CPU utilization.

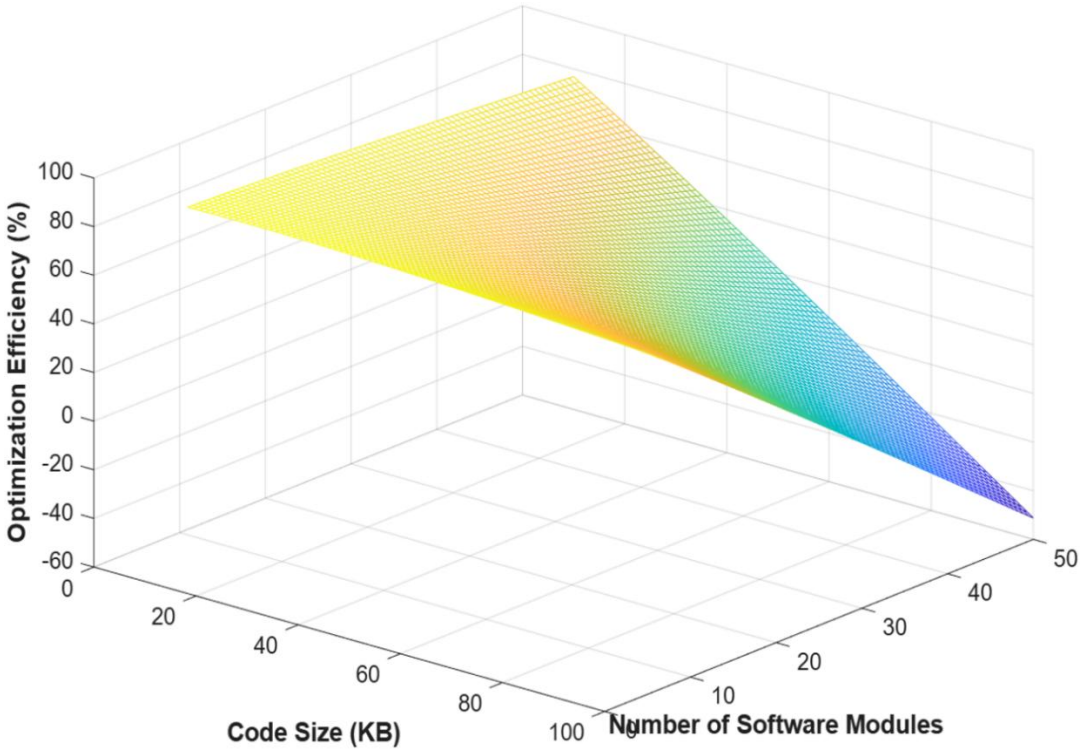


Fig. 6. Scalability with increasing code size.

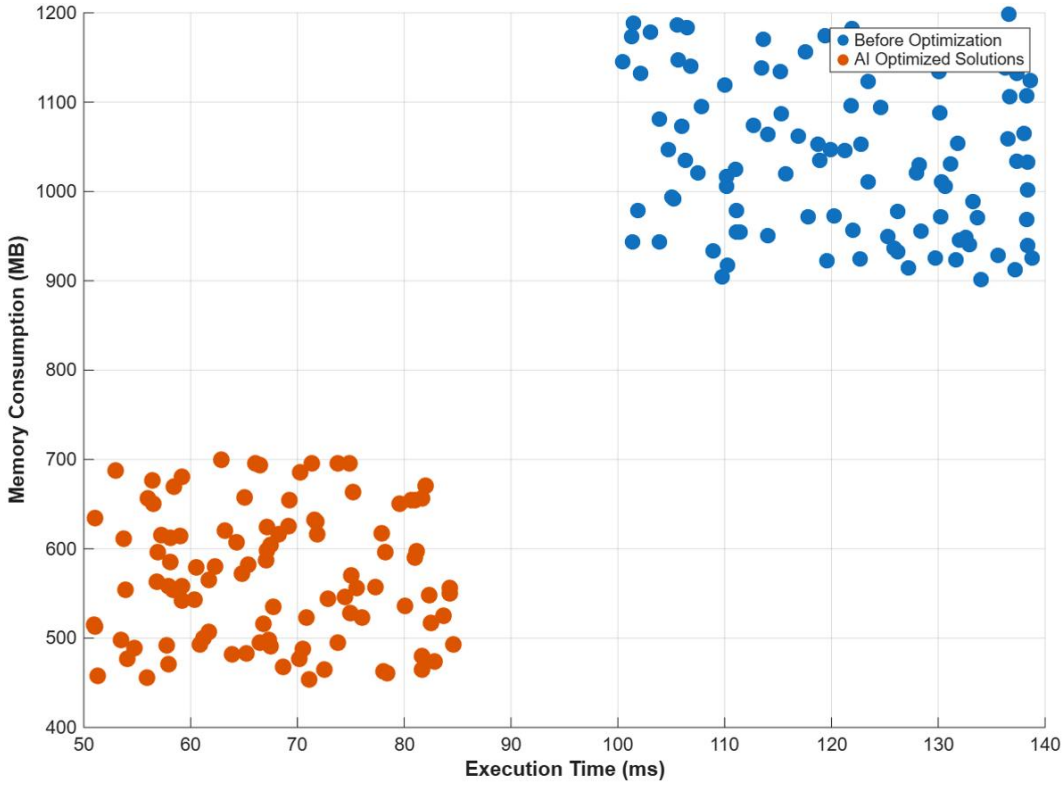


Fig. 7. Execution time vs memory usage (before vs after optimization).

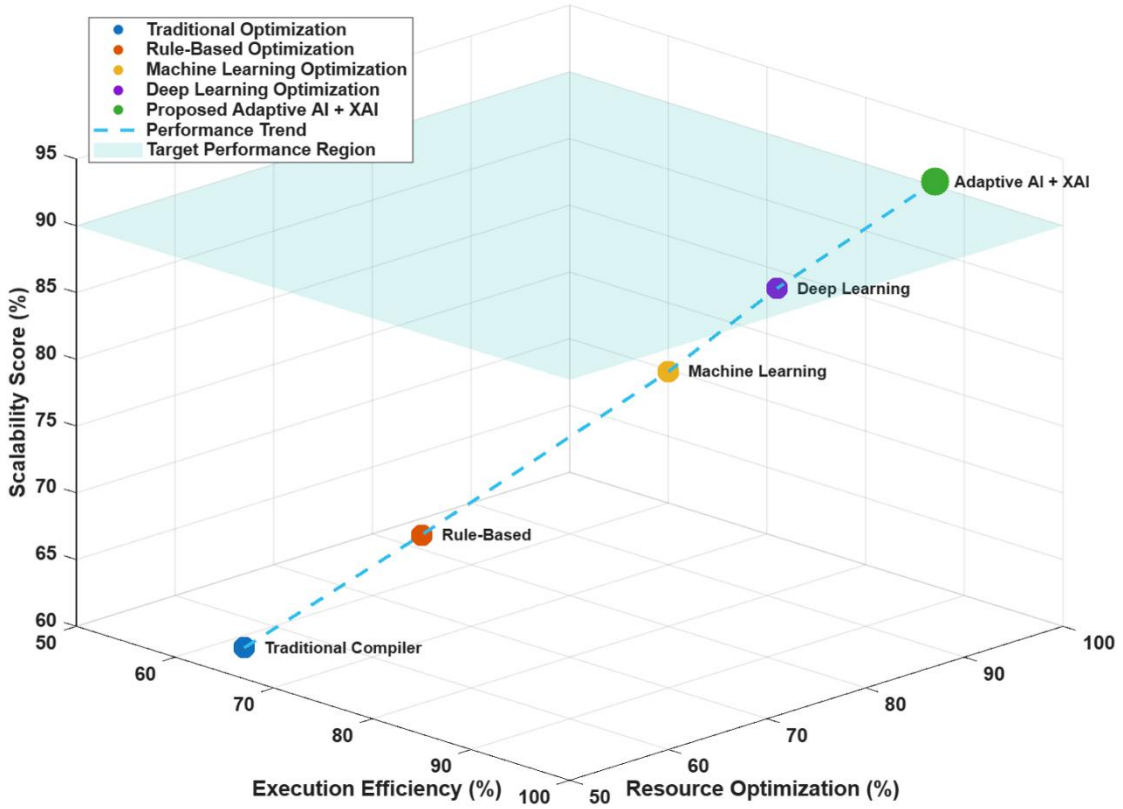


Fig. 8. Performance comparison: Traditional optimization vs AI-based optimization.

TABLE III. PERFORMANCE EVALUATION OF THE PROPOSED ADAPTIVE AI-DRIVEN OPTIMIZATION FRAMEWORK

Application Type	Baseline Execution Time (s)	Optimized Execution Time (s)	Execution Time Improvement (%)	Memory Usage Before (MB)	Memory Usage After (MB)	Memory Reduction (%)
Image Processing	100	80	20.0	500	400	20.0
Sorting Algorithm	100	80	20.0	550	465	15.5
Database Query	100	76	24.0	650	570	12.3
Machine Learning	100	73	27.0	950	850	10.5
Web Server	100	68	32.0	1100	950	13.6
Average Performance Gain	100	75.4	24.6	750	647	14.4

The scalability analysis shown in Fig. 6 reveals that the framework has the capability to scale with code size. Although the classic-based optimization methods have a tendency to incur diminishing returns, the given AI-based system progressively improves in terms of execution time, despite a correspondingly rising code size. This means that the framework can support large-scale applications, and both small and large-sized datasets are optimized. The AI model adjusts to the code size increase and changes optimization tactics, which provide performance improvements with scale. Table III presents the performance evaluation of the proposed adaptive AI-driven optimization framework across different application domains. The results demonstrate improvements in execution time and memory consumption, with an average execution time reduction of 24.6% and memory reduction of 14.4%. The framework achieves the highest performance improvement for web server and machine learning applications due to effective bottleneck identification and adaptive optimization strategy selection.

Fig. 7 and Fig. 8 have the advantage of developers in terms of explainability. Functionalities of the AI model that allow explaining why specific optimizations are selected offer a clear understanding of why some are selected. Fig. 7 would compare the execution time and memory consumption prior to the optimization and after the optimization so that the developers would be able to see the physical results of the modifications. In Fig. 8, a comparison between the traditional and the AI-based framework can be done and this clearly visualizes the increase in performance. The explainable optimization mapping method shows the developers how the decisions were made, and this increases confidence in the framework. This openness is essential to adoption, particularly with the production of any type of code where the developer must guarantee data integrity and reliability of the optimized code.

Table IV compares the proposed framework with existing optimization approaches based on execution efficiency, memory

optimization, CPU utilization reduction, and adaptability. The proposed ML and XAI-based framework achieves superior performance with a 30% execution time reduction, 25% memory reduction, and 15% CPU utilization improvement. The results confirm that integrating adaptive learning and explainable decision-making provides better resource-aware optimization than conventional approaches.

Table V presents the scalability evaluation of the proposed framework under varying code sizes and complexity levels. The results demonstrate that the framework maintains consistent optimization performance as software complexity increases, achieving up to 28% improvement in execution efficiency for larger codebases. The improvement in optimization prediction accuracy and the availability of interpretable decision insights further confirm the effectiveness of integrating Machine Learning with Explainable AI for scalable code optimization. Despite these encouraging results, some limitations were identified during the experimental evaluation. The performance of the AI-driven optimization framework is highly influenced by the quality, diversity, and representativeness of the training data used for model development. In cases where the characteristics of target applications significantly differ from the training samples, the selected optimization strategies may not achieve the expected performance improvements. Therefore, expanding the training dataset with diverse software patterns and hardware configurations is necessary to improve the generalization capability of the framework.

TABLE IV. RESOURCE UTILIZATION AND COMPUTATIONAL EFFICIENCY COMPARISON

Optimization Method	Execution Time Reduction (%)	Memory Reduction (%)	CPU Utilization Reduction (%)	Resource Efficiency Score (%)	Optimization Adaptability
Traditional Compiler Optimization	12.5	8.0	6.5	9.0	Limited
Rule-Based Optimization	18.0	11.5	9.0	12.8	Moderate
Machine Learning-Based Optimization	24.0	18.0	12.5	18.2	High
Deep Learning-Based Optimization	27.0	21.0	14.0	21.5	High
Proposed Adaptive AI + XAI Framework	30.0	25.0	15.0	23.3	Very High

TABLE V. SCALABILITY AND AI OPTIMIZATION CAPABILITY EVALUATION

Code Size (KB)	Traditional Optimization Execution Time (s)	Proposed AI Optimization Execution Time (s)	Performance Improvement (%)	Optimization Prediction Accuracy (%)	XAI Decision Transparency
10	100	80	20.0	88	Medium
20	200	150	25.0	90	High
30	300	220	26.7	92	High
40	400	290	27.5	93	High
50	500	360	28.0	94	Very High
Average	300	220	25.4	91.4	-

Also, although this framework has high scalability capabilities with most applications, performance improvements might begin to level off with large-scale systems or applications that have complicated non-standard architecture. More studies need to be done to upgrade the AI models in order to deal with such cases in a better way. Finally, the computational cost of XAI (Explainable AI) may become complex, but in many cases, it is offset by the advantages of interpretability.

VIII. CONCLUSION

The proposed adaptive hardware-aware AI-driven code optimization framework demonstrates effective improvements in embedded computing performance through the integration of Machine Learning and Explainable AI techniques. The proposed framework integrates software-level characteristics with hardware resource parameters to identify optimization opportunities and determine appropriate optimization strategies. Experimental results demonstrate improvements in embedded system performance, achieving up to 30% reduction in execution time, 25% reduction in memory consumption, and approximately a 15% reduction in CPU utilization, thereby confirming the effectiveness of the framework in enhancing computational efficiency and enabling resource-aware optimization. The scalability evaluation further shows that the framework maintains consistent optimization performance as code complexity increases, supporting its applicability in resource-constrained embedded environments. Moreover, the integration of Explainable AI improves the transparency of optimization decisions by providing insights into the contribution of software and hardware features, allowing developers to better understand and trust AI-assisted optimization outcomes. However, the framework's performance remains dependent on training data quality and may vary across different embedded architectures and hardware configurations. Future research will focus on extending the framework for heterogeneous embedded platforms, incorporating additional hardware-aware parameters such as energy consumption and cache utilization, and developing lightweight AI models for real-time deployment. These enhancements will further improve the adaptability, efficiency, and practical applicability of AI-driven optimization solutions for next-generation embedded computing systems.

REFERENCES

[1] T. Bosse et al., "A Standard Framework for AI-Driven Optimizations in Various Complex Domains," 2024 Winter Simulation Conference (WSC), Orlando, FL, USA, 2024, pp. 1966-1977

[2] L. Huang, H. Zhang, R. Li, Y. Ge and J. Wang, "AI Coding: Learning to Construct Error Correction Codes," in IEEE Transactions on Communications, vol. 68, no. 1, pp. 26-39, Jan. 2020

[3] Y. Peng, A. D. Gotmare, M. R. Lyu, C. Xiong, S. Savarese and D. Sahoo, "PerfCodeGen: Improving Performance of LLM-Generated Code with Execution Feedback," 2025 IEEE/ACM Second International Conference on AI Foundation Models and Software Engineering (Forge), Ottawa, ON, Canada, 2025, pp. 1-13

[4] J. Vuppalapati and S. Kedari, "Stratification of, albeit Mathematical Optimization and Artificial Intelligent (AI) Driven, High-Risk Elderly Outpatients for priority house call visits - a framework to transform healthcare services from reactive to preventive," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 4955-4960

[5] Z. Feng, H. Wang, Z. Lin, R. Ma and D. Niyato, "Coupled Phase-Amplitude RIS for Secure SCMA in Cognitive Satellite-Terrestrial Networks: An MADRL Optimization Framework," in IEEE Internet of Things Journal, vol. 13, no. 1, pp. 284-295, 1 Jan.1, 2026

[6] N. Arshad, T. A. Butt and M. Iqbal, "A Comprehensive Framework for Intelligent, Scalable, and Performance-Optimized Software Development," in IEEE Access, vol. 13, pp. 74062-74077, 2025

[7] V. Rajendran, D. Besiahgari, S. C. Patil, M. Chandrashekaraiah and V. Challagulla, "A Multi-Agent LLM Environment for Software Design and Refactoring: A Conceptual Framework," SoutheastCon 2025, Concord, NC, USA, 2025, pp. 488-493

[8] S. Rajawat, S. B. Goyal and M. N. H. Ibrahim, "AI-Driven Compression and Redundancy Removal Techniques for Real-Time Communication in Healthcare Networks," 2024 IEEE Globecom Workshops (GC Wkshps), Cape Town, South Africa, 2024, pp. 1-6

[9] Dhole, Sampada Abhijit , Patil, Jayamala Kumar & Singh, Shivoham (2025) A novel cryptographic approach to evaluate the security of federated learning frameworks, Journal of Discrete Mathematical Sciences and Cryptography, 28:5-A, 1527–1536

[10] S. T. Kim, Y. K. Kim, J. W. Park, K. S. Kim and T. H. Kim, "Evolution and Applications of Codeless AI Model Generation Tools for AI-Driven Education," 2025 World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC), Daegu, Korea, Republic of, 2025, pp. 1-4

[11] Y. Zhang, Z. Duan, Y. Huang and F. Zhu, "Low-Rank Adaptation of Pre-Trained Vision Backbones for Energy-Efficient Image Coding For Machines," 2025 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2025, pp. 1163-1168

[12] Dekka, Satish, Thiruvenkadam, T. & Prasad, K. D. V.(2024) Cryptographic image-based data security strategies in wireless sensor networks, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-A, 293–304

[13] S. S. Poojary, S. Kamble and G. S. Mamatha, "A Scalable DevOps Framework Using Kubernetes-Orchestrated Microservices for Cloud-Native Infrastructure," 2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2025, pp. 1-6

[14] C. Maddilety, G. Lokesh, A. V. Vathsala, M. Swapna, K. Samunnisa and M. Bhavsingh, "AI Driven Self Healing Software Architectures for Automated Code Diagnostics and Repair," 2025 International Conference on Sustainable Communication Networks and Application (ICSCN), Theni, India, 2025, pp. 1728-1735

[15] Sarma, Parismita & Rahman, Mirzanur (2024) Mathematical analysis of wavelet-based multi-image compression in medical diagnostics, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-B, 675–687

[16] Gupta, M. Sharma and A. Srivastava, "A Novel Dimensionality Reduction-based Software Bug Prediction using a Bat-Inspired Algorithm," 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2023, pp. 278-285

[17] L. Ruggahakotuwa, L. Rupasinghe and P. Abeygunawardhana, "Code Vulnerability Identification and Code Improvement using Advanced Machine Learning," 2019 International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2019, pp. 186-191

[18] J. Lu, F. Li, C. Liu, L. Li, X. Feng and J. Xue, "CloudRaid: Detecting Distributed Concurrency Bugs via Log Mining and Enhancement," in IEEE Transactions on Software Engineering, vol. 48, no. 2, pp. 662-677, 1 Feb. 2022

[19] V. Rastogi and A. Pandey, "Code Analysis to Prioritize and Fix Bugs Using Top-K Pruning Approach with help of Machine Learning," 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 1476-1482

[20] K. Aldi, R. Septianto, R. Yulianto and R. Taskia, "Multi-Tool Static Code Analysis for Code Smell and Bug Detection: A Case Study on the Java-Design-Patterns Repository," 2025 International Conference on Information Technology Research and Innovation (ICITRI), Jakarta, Indonesia, 2025, pp. 1-6

[21] H. Sajnani, V. Saini and C. V. Lopes, "A Comparative Study of Bug Patterns in Java Cloned and Non-cloned Code," 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation, Victoria, BC, Canada, 2014, pp. 21-30

Enhanced Distributed Bandwidth Allocation Algorithm for Network Latency Reduction in IEEE 802.17 Rings

Fahd Alharbi
Electrical Engineering Department-Faculty of Engineering, Rabigh, King Abdulaziz University, Jeddah 21589, Saudi Arabia

***Abstract*—The IEEE 802.17 Resilient Packet Ring (RPR) technology is a cornerstone for the rapid expansion of metropolitan digital infrastructure. It is recognized for its robustness and spatial reuse capabilities. However, under highly dynamic and asymmetric traffic loads, standard RPR fairness algorithms often suffer from bandwidth oscillations, leading to unpredictable queuing delays and increased network latency. To address this limitation, this study introduces an Enhanced Distributed Bandwidth Allocation (EDBA) algorithm designed specifically for network latency reduction. By incorporating a filtered transit queue length into the fair rate calculation, EDBA dynamically calibrates bandwidth allocation at the source nodes before congestion propagates across the transit paths. Performance evaluations via network simulations demonstrate that the proposed algorithm significantly reduces average packet latency by up to 94%, while maintaining fairness and high throughput across the ring topology.**

***Keywords*—RPR; IEEE 802.17; bandwidth; latency; fairness**

I. Introduction

Smart cities handle the challenges of urban growth through technological integration, thereby enhancing the quality of life, upgrading security measures, and maximizing resource efficiency [1]. Smart environments are experiencing a rapid proliferation of time-critical applications. These include intelligent transportation systems (ITS), surveillance systems, real-time video streaming, grid management, financial trading, and telemedicine [2]. With the rise of real-time applications that necessitate rapid and efficient data transmission, latency is a critical performance metric in optical networks. Fundamentally, latency represents the time delay experienced between data transmission and reception. Within optical networks, this total delay comprises several components, including propagation delay through the fiber, queuing delay, and processing time at individual network nodes [3]. Metropolitan area networks (MANs) widely adopt ring aggregation architectures to ensure high availability and robust fault tolerance [4]. Resilient Packet Ring (RPR), standardized under IEEE 802.17, is a specialized ring topology designed primarily for Metropolitan Area Networks (MANs) and Wide Area Networks (WANs). The defining feature of RPR is its exceptional fault tolerance, allowing uninterrupted data transmission even during severe link failures. This robust resilience has driven its widespread adoption among MAN and WAN service providers, who traditionally rely on legacy systems like Synchronous Optical Network (SONET) or Synchronous Digital Hierarchy (SDH) architectures [5].

The Resilient Packet Ring (RPR) architecture (Fig. 1) employs dual counter-rotating optical rings to ensure robust protection and high bandwidth utilization. Each RPR node includes dedicated buffers for both local and transit traffic (Fig. 2). The primary buffer manages high-quality traffic requiring guaranteed data rates and low jitter, whereas the secondary buffer handles low-quality traffic regulated by a fairness algorithm. The primary buffer has priority to be served first, then the local buffer, as long as the secondary buffer length is less than a threshold of 1/4 of the buffer size. If the secondary buffer length exceeds the threshold, it will have a priority over the local buffer. The fairness algorithm operates via two distinct mechanisms: the aggressive method, which triggers congestion control when the secondary buffer exceeds a specific threshold, and the conservative method, which detects congestion when the combined local and transit traffic exceeds the available bandwidth for low-priority data [6].

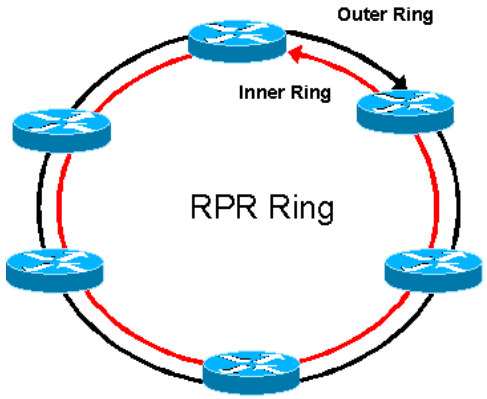


Fig. 1. The packet ring.

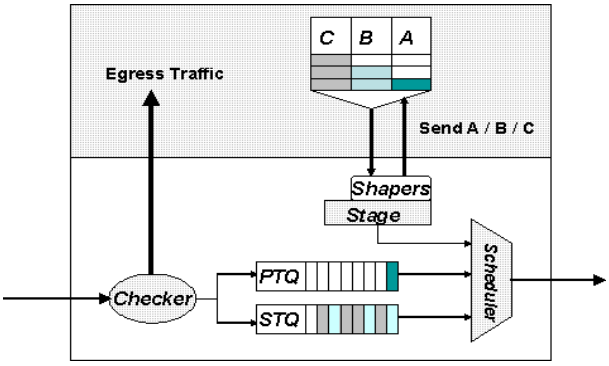


Fig. 2. RPR node architecture.

While RPR preserves the sub-50ms protection of traditional SONET/SDH rings and enhances efficiency through spatial reuse, its standard distributed fairness mechanism degrades under bursty, non-uniform traffic. Intense congestion and bandwidth competition among nodes cause fairness algorithms to oscillate severely. Consequently, this introduces heavy queuing jitter and severe packet latency bottlenecks [7].

To stabilize bandwidth allocation and mitigate these oscillations, several alternative algorithms have been proposed. For instance, the DVSR algorithm [8] monitors network nodes' demands and applies max-min fairness to eliminate oscillations, though it requires a computational complexity of *O (n log n)*, where n is the number of data flows. Conversely, the Distributed Bandwidth Allocation (DBA) algorithm [9] achieves fairness with a lower complexity of *O (1)*. Control theory has also been applied to optimize fair rate calculations. Researchers have utilized a classical PID controller to stabilize the transit buffer and regulate the bandwidth allocation process [10]. On the other hand, a fuzzy logic controller was designed to monitor the transit buffer, then increase or decrease the fair rate to maintain stability and achieve fairness [11]. Also, a combination of a Smith predictor, P-controller, and fuzzy tuning method was introduced in [6] to handle packet rings with large propagation delays. While these methods regulate bandwidth either by matching injected traffic to available capacity via DVSR and DBA algorithms or stabilizing the low-priority buffer at a target level utilizing PID and fuzzy controllers, transit traffic still accumulates in low-priority buffers during the convergence phase. This accumulation increases queuing delay and exacerbates network latency.

While the expansion of smart city frameworks requires low-latency data processing and optimal resource utilization, traditional bandwidth allocation methods introduce critical bottlenecks [12]. To address these limitations, the Distributed Bandwidth Allocation (DBA) algorithm [9] is leveraged due to its operational efficiency and low computational complexity. The objective is to design an Enhanced Distributed Bandwidth Allocation (EDBA) algorithm. By optimizing the rate-throttling process, EDBA ensures rapid latency reduction without sacrificing the foundational throughput and fairness guarantees of the IEEE 802.17 standard.

II. EDBA ALGORITHM

In this section, we review the Distributed Bandwidth Allocation (DBA) algorithm. To simplify the analysis without loss of generality, we consider a single counter-rotating path: the outer ring comprising *N* nodes, indexed sequentially along the traffic direction. Let *C* denote the available capacity allocated for low-priority traffic. The aggregate arrival rate at link *k* during the *nth* regulation interval is defined as follows:

$$A(n) = \sum_{i=1}^{M} R_i \quad (1)$$

where, R_i is the amount of data transmitted from node *i* and passing through node *k,* and *M* is the number of flows. The goal of the fairness algorithm is to maximize the fair rate F_k subject to the following conditions:

$$A \leq C \quad (2)$$

$$R_i \leq F_k \text{ , for all flows going through link } k \quad (3)$$

The algorithm uses the following equation to estimate the fair rate:

$$F_k(n) = F_k(n-1) + [C - A(n)]/M \quad (4)$$

The primary objective of the fair rate estimation equation [Eq. (4)] is to adjust *F* such that the aggregate arrival rate *A* matches the available capacity *C*. Analytical models and experimental results have established that the DBA algorithm converges to the fair rate rapidly [9]. However, as demonstrated in the subsequent section, our simulation experiments reveal that the transit buffer at the congestion point accumulates significantly during this estimation period. Because the DBA algorithm lacks a dedicated queue-draining mechanism, passing packets experience long queuing delays. Consequently, while the DBA algorithm successfully regulates bandwidth allocation, it does so at the expense of increased network latency. This structural limitation reduces the algorithm's ability to guarantee the strict Quality of Service (QoS) requirements of time-critical smart city applications; a deficiency that this work directly addresses.

The objectives of the proposed method are to regulate the fair rate and minimize the packet delay. Accordingly, the buffer length is incorporated in the fair rate calculation equation and is rewritten as follows:

$$F_k(n) = F_k(n-1) + \left[C - A(n) - \beta Q_f(n)\right]/M \quad (5)$$

where, Q_f is the filtered version of the low-priority buffer length, and the parameter β controls the speed of clearing the buffer. The filtered buffer length is updated as follows:

$$Q_f(n) = (1-\alpha)Q_f(n-1) + \alpha\, Q_i(n) \quad (6)$$

where, Q_i is the instantaneous value of the low-priority buffer length, and the filter coefficient is $\alpha \in [0,1]$ [13].

III. RESULTS AND DISCUSSION

In this section, we conduct two simulation experiments. The first experiment is the simple congestion points scenario shown in Fig. 3, where all links have the same capacity of 622 Mbps, and each link has a propagation delay of 0.1 ms. All data flows are greedy and start at time 0. The measurement time interval is set to T=1ms. The experiment is conducted using the RPR simulator [14].

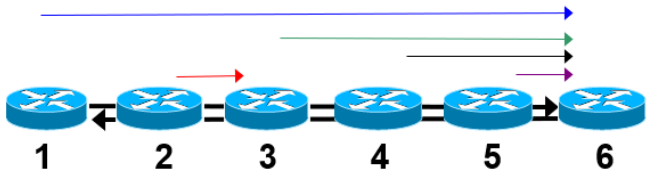


Fig. 3. Simple congestion points scenario.

In this scenario, there are two congestion points. The first congestion point is between switches 5 and 6, where four flows compete for the bandwidth, and a fair share is 155 Mbps. The second point is between switches 2 and 3, where the competing flows are flow (1,6) and flow (2,3). Since flow (1,6) would be limited to the fair share reported by switch 5, flow (2,3) should be able to utilize the remaining bandwidth on link 2-3 and transmit at a rate of 467Mbps. The performance of the

proposed EDBA algorithm is evaluated against two different algorithms. The DBA algorithm is considered as a rate-based algorithm, while the PID controller is considered a queue-based algorithm. The evaluation focuses on the ability to achieve fairness and high resource utilization and also reduce the network latency. The performance of the DBA algorithm, the PID controller, and the proposed EDBA in regulating the bandwidth allocation are shown in Fig. 4, Fig. 5, and Fig. 6, respectively. All algorithms successfully converge very quickly, enabling all flows to achieve their fair shares.

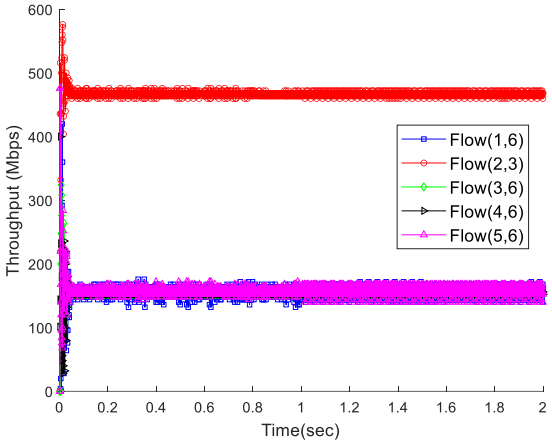


Fig. 4. Simple congestion points scenario: flows convergence (DBA).

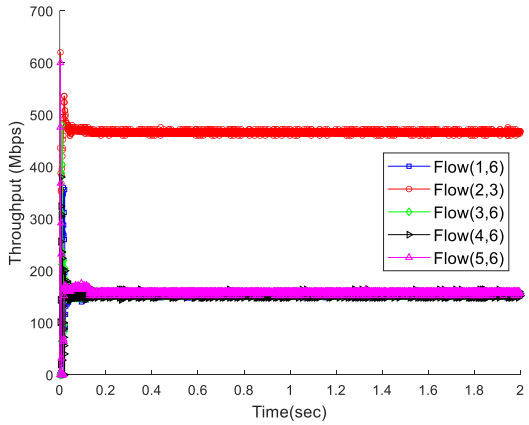


Fig. 5. Simple congestion points scenario: flows convergence (PID).

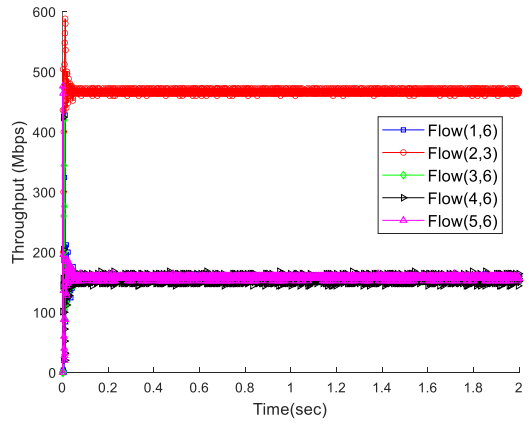


Fig. 6. Simple congestion points scenario: flows convergence (EDBA).

Due to the simultaneous congestion at switches 2 and 5, the low-priority transit buffer occupancy at these nodes accumulates significantly, reaching approximately 25% of the total buffer capacity during the convergence period, as illustrated in Fig. 7. These results demonstrate that the DBA algorithm cannot clear these queues once accumulation occurs.

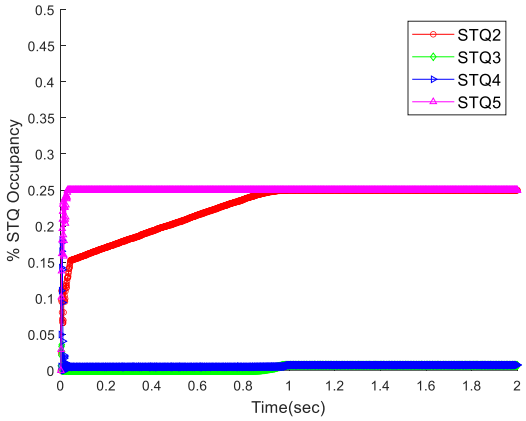


Fig. 7. Simple congestion points scenario: buffer length (DBA).

On the other hand, maintaining a stable queue at a pre-defined target level is a primary objective of the PID controller. Consequently, as illustrated in Fig. 8, the low-priority buffer occupancy at the bottleneck points stabilizes at the target threshold of one-sixteenth (6.25%) of the total buffer capacity. In contrast, the proposed EDBA algorithm successfully empties the accumulated traffic from the low-priority transit buffers, as demonstrated in Fig. 9. By dynamically integrating the filtered buffer occupancy metric Q_f, into the modified fair rate estimation [Eq. (5)], EDBA is capable of clearing the ring transit queues while fully preserving the throughput and fairness mandates of the IEEE 802.17 standard.

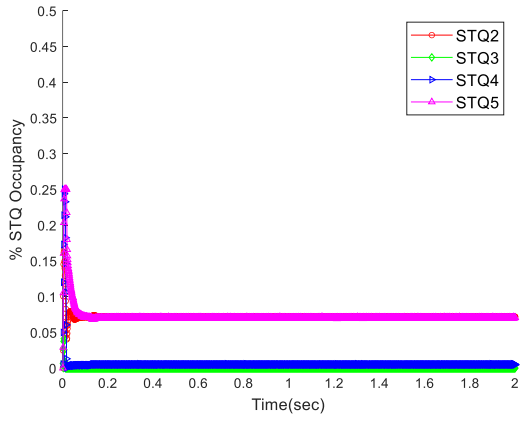


Fig. 8. Simple congestion points scenario: buffer length (PID).

For all algorithms, we compare the delay of packets transmitted from switch 1 to switch 6, where they pass through two congestion points. Fig. 10 illustrates the transient and packet network latency curves for the three evaluated algorithms. As expected from the buffer occupancy behaviors, the DBA algorithm exhibits a severe, continuous growth in network delay during the transient convergence period, ultimately saturating at a high steady-state latency of

approximately 18.124 ms. This rapid, continuous increase confirms the accumulation of low-priority traffic within the transit buffers; without a dedicated queue-clearing mechanism, packets are subjected to persistent, cumulative queuing delays.

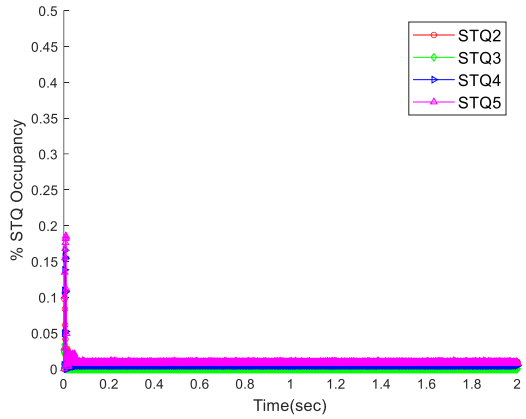


Fig. 9. Simple congestion points scenario: buffer length (EDBA).

In contrast, the queue-based PID controller successfully stabilizes the latency curve, reducing the network delay to roughly 5.383 ms. Because the PID algorithm regulates the transit buffer to a specific target length, it prevents the unbounded delay increase observed in the DBA algorithm. However, it still maintains a permanent, non-zero queuing overhead to enforce its operating setpoint.

Remarkably, the proposed EDBA algorithm outperforms both the DBA algorithm and the PID controller. The EDBA network latency curve drops quickly and stabilizes at approximately 1.104 ms. By emptying the low-priority transit queues using the filtered buffer length variable, Q_f , EDBA almost eliminates queuing delays. This represents an approximate 94% latency reduction compared to DBA and a 79% reduction compared to the PID controller, demonstrating EDBA's capability to sustain the strict low-latency requirements demanded by time-critical smart city applications.

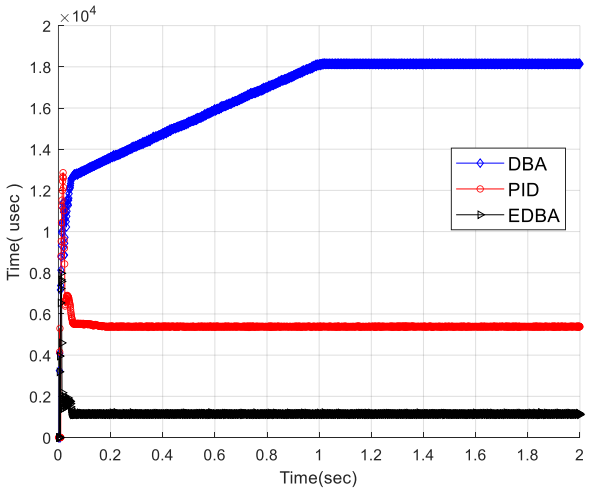


Fig. 10. The transient and packet network latency curves.

In the second experiment (Fig. 11), all flows are greedy and start at time 0. In this scenario, data packets transmitted from Switch 1 to Switch 6 pass through multiple congestion points. The fair-share bandwidth at each congestion point is 311 Mbps because each point is shared by only two flows.

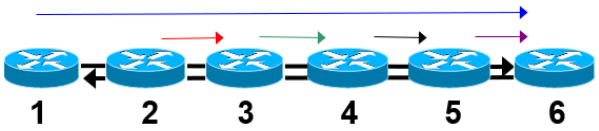


Fig. 11. Multiple congestion points scenario.

Similarly, the DBA algorithm, PID controller, and proposed EDBA algorithm successfully regulate the bandwidth allocation. As depicted in Fig. 12–14, each flow consistently achieves a bandwidth of 311 Mbps.

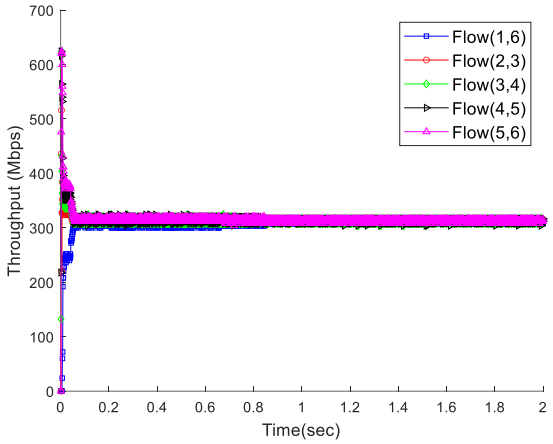


Fig. 12. Multiple congestion points scenario: flows convergence (DBA).

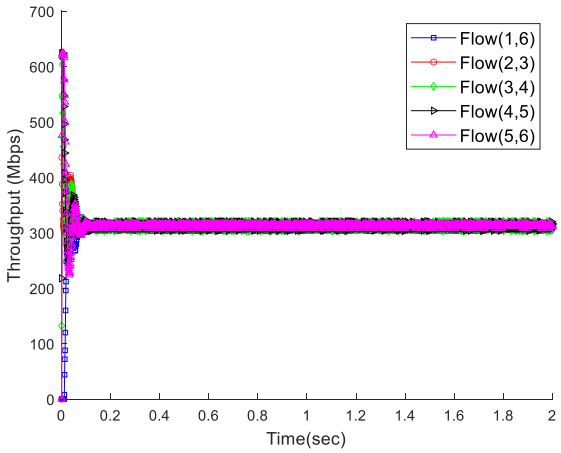


Fig. 13. Multiple congestion points scenario: flows convergence (PID).

This scenario introduces four simultaneous congestion points at switches 2 through 5. As illustrated in Fig. 15, the low-priority transit buffer occupancy at these nodes builds up rapidly, reaching roughly 25% of buffer capacity during convergence. This highlights a key limitation: the DBA algorithm lacks the capability to drain these queues once accumulation takes place.

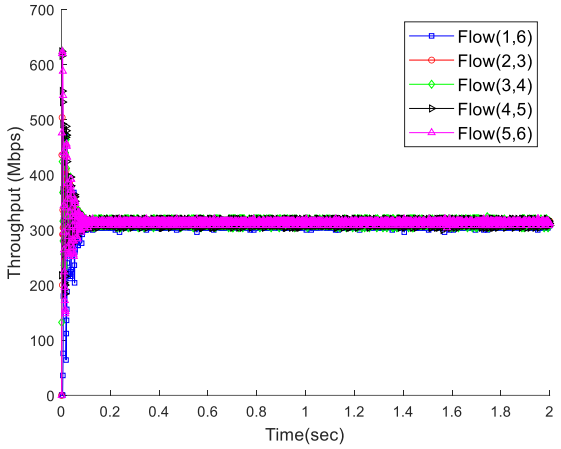


Fig. 14. Multiple congestion points scenario: flows convergence (EDBA).

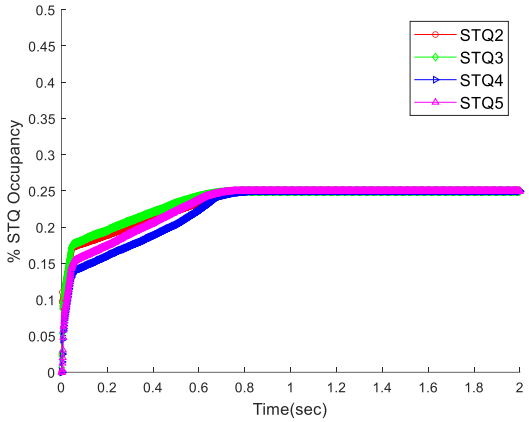


Fig. 15. Multiple congestion points scenario: buffer length (DBA).

While the PID controller focuses primarily on stabilizing the queue at a predefined target level, thereby holding the low-priority buffer occupancy at the bottleneck points to 6.25% of the buffer capacity (Fig. 16), the proposed EDBA algorithm successfully evacuates the accumulated traffic (Fig. 17). This is achieved by dynamically integrating the filtered buffer occupancy measurement Q_f , into the fair rate estimation equation [Eq. (5)]. Consequently, EDBA clears the ring transit queues without compromising the throughput and fairness mandates of the IEEE 802.17 standard.

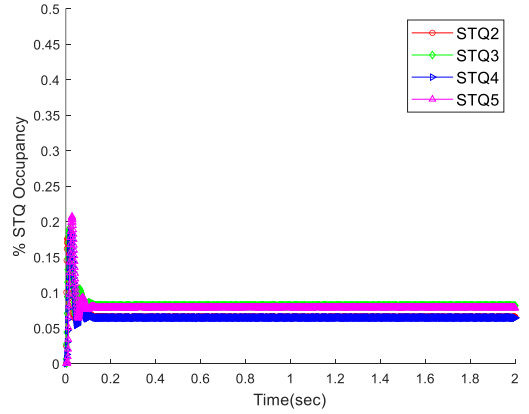


Fig. 16. Multiple congestion points scenario: buffer length (PID).

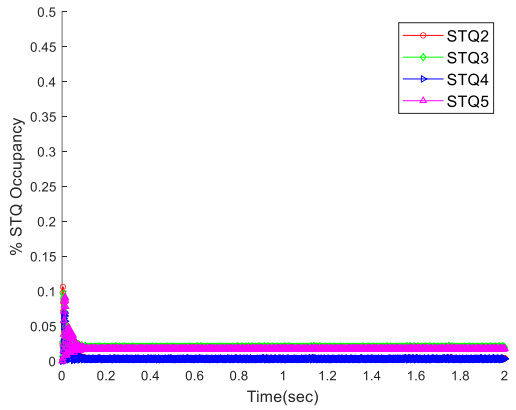


Fig. 17. Multiple congestion points scenario: buffer length (EDBA)

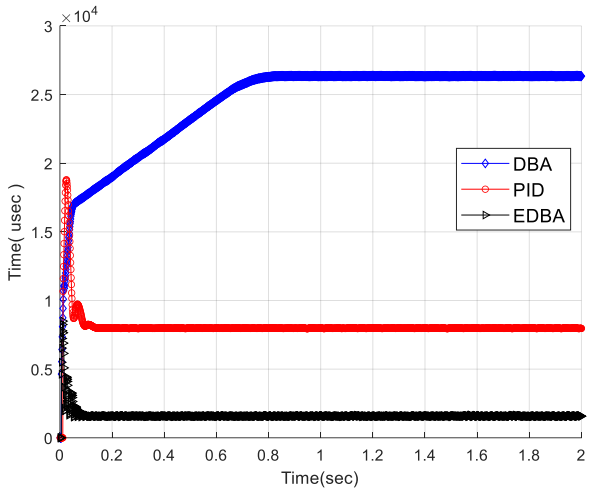


Fig. 18. Flow (1,6) packet delay.

Fig. 18 illustrates the network latency performance for the three evaluated algorithms under the multi-congestion scenario, specifically showing the delay of Flow (1,6) packets transmitted from switch 1 to switch 6. Because these data packets pass through multiple concurrent traffic bottlenecks, the DBA algorithm exhibits a continuous increase in latency during the transient phase, reaching a delay of approximately 26.34 ms. Furthermore, since the rate-based DBA algorithm lacks a queue-draining mechanism, the packet accumulation that builds up in the transit buffers during the fair-rate estimation period exposes all passing packets to high latency.

The queue-based PID controller successfully mitigates network latency growth. At the beginning of congestion, the PID controller experiences a sharp, transient overshoot; however, it rapidly reduces the latency within a few milliseconds, settling at a delay of approximately 7.965 ms. While effective at reducing network delay, the PID controller must maintain its transit queues at a non-zero setpoint to operate, thereby leaving a permanent queuing overhead in the network path

The proposed EDBA algorithm outperforms both approaches. Upon the initialization of concurrent flows, EDBA effectively limits the transient delay spike. Driven by the filtered buffer occupancy metric, Q_f , the rate-throttling

mechanism immediately drains the transient queue accumulation. Consequently, the latency curve decays sharply and stabilizes rapidly, maintaining a remarkably low delay of approximately 1.571 ms.

This optimized behavior translates to substantial performance improvements for delay-sensitive network infrastructures: a 94.0% latency reduction compared to the rate-based DBA algorithm and an 80.0% latency reduction compared to the queue-stabilizing PID controller. By completely evacuating the transient queues rather than containing them, the EDBA algorithm proves highly effective at sustaining the strict, low-latency Quality of Service (QoS) required by next-generation smart city applications.

IV. CONCLUSION

This study presented the Enhanced Dynamic Bandwidth Allocation (EDBA) algorithm, a novel queue-aware framework designed to mitigate transient congestion and optimize latency performance in IEEE 802.17 Resilient Packet Ring (RPR) networks. Experimental evaluations under concurrent multi-congestion scenarios demonstrated that while the conventional rate-based DBA algorithm fails to drain accumulated buffers and the PID controller incurs a permanent queuing overhead, the proposed EDBA algorithm completely evacuates transient queues. Specifically, EDBA maintains a stable, low steady-state delay of approximately 1.571 ms, yielding a 94.0% latency reduction relative to the DBA algorithm and an 80.0% reduction compared to the PID controller. By successfully balancing transient queue evacuation with the strict throughput and fairness mandates of the IEEE 802.17 standard, EDBA proves to be a highly effective solution for sustaining the low-latency Quality of Service (QoS) required by next-generation smart city applications.

While the proposed EDBA algorithm achieves exceptional performance using a static filtering configuration, adaptive parameter tuning remains an avenue for future exploration. Subsequent research will focus on developing an adaptive tuning mechanism for the buffer clearing parameter, β, and the buffer filter coefficient, α. Dynamically adjusting these parameters based on real-time network dynamics will allow the algorithm to maintain optimal responsiveness across varying traffic intensities.

REFERENCES

[1] Z. E. Ahmed, A. A. Hashim, R. A. Saeed and M. M. Saeed, "Enhancing Smart City Mobility Using Software Defined Networks," *9th International Conference on Mechatronics Engineering (ICOM)*, Kuala Lumpur, Malaysia, 2024, pp. 299-303.

[2] Patel, C. M., "Edge Computing for Low-Latency loT Applications in Smart Cities," *Smart Internet of Things*, *1*(4), 2024, pp.282-288.

[3] "Latency in Optical Networks:How It Impacts Real-Time Applications," *Patsnap.com*,2025. https://eureka.patsnap.com/article/latency-in-optical-networks-how-it-impacts-real-time-applications (accessed Aug. 14, 2026).

[4] Y. Nakayama and N. Oota, "N Rate N+1 Color Marking: Per-Flow Fairness in Ring Aggregation Networks," in *IEEE Transactions on Communications*, vol. 62, no. 12, pp. 4401-4412, Dec. 2014

[5] Minhas T, Khan S, Arslan F, Ali A, Hussain A, Ali J. A Versatile Resilience Packet Ring Protocol Model for Homogeneous Networks. *Applied Sciences*. 2023; 13(8):4660.

[6] Fahd Alharbi, "Adaptive Predictive Proportional Controller for IEEE 802.17," *IEEE Access*, vol. 8, pp. 23082-23089, 2020.

[7] M. M. Iqbal, M. D. Awan, M. Sikander, R. A. Bakar, "VRPR: A New Data Center Protocol for Enhanced Network Performance, Resilience and Recovery," *IEEE Access*, vol. 12, pp. 130524–130545, 2024.

[8] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, E. Knightly, "Design, Analysis, and Implementation of DVSR: A Fair, High Performance Protocol for Packet Rings," *IEEE/ACM Transactions on Networking*, 12: pp. 85-102, 2024.

[9] Fahd Alharbi and Nirwan Ansari, "Distributed bandwidth allocation for resilient packet ring networks," *Computer Networks*, vol. 49, pp. 161-171, 2005.

[10] Tan, Liansheng et al. "PID-RPR: A High Performance Bandwidth Allocation Approach for RPR Networks." *IEICE Trans. Commun.* 88-B (2005): 2872-2878.

[11] W. -S. Tang, W. -C. Wang, C. -J. Chang, Y. -T. Hsieh, "An effective fair bandwidth allocation in resilient packet ring," *5th International ICST Conference on Communications and Networking in China,* Beijing, China, pp. 1-5, 2010.

[12] R. Fredyan, L. H. Pramono, and . Khairunnisa, "Adaptive Dynamic Bandwidth Allocation in Smart Cities Using Software-Defined Networking", *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 24957–24963, Aug. 2025.

[13] Jiang, F.C., Feng, C.W., Zhu, C., Sun, Y.," Performance analysis of active queue management algorithm based on reinforcement learning,". Journal Européen des Systèmes Automatisés, Vol. 53, No. 5, (2020, pp. 637-644.

[14] S. Gjessing, "The Simula RPR Simulator implemented in Java," *Simula Research Laboratory Technical Report* 2003-12.

Hybrid Resource Allocation Technique Based on Double Q Learning and Metaheuristic Algorithms in Optimizing the Allocation of Tasks in Cloud Computing

V. Mahalakshmi[1], S. Poornima[2]
Research Scholar, Department of Computer Science, Vel's Institute of Science,
Technology and Advanced Studies, Chennai, Tamil Nadu, India[1]
Department of Computer Science, Vels Institute of Science Technology and Advanced Studies (VISTAS),
Chennai, Tamil Nadu, India[2]

***Abstract*—Cloud computing (CC) has grown as a core component of modern computing facilities, providing users with adaptable, on-demand services. In dynamic environments, the effective utilization of resources is significant for performance, cost-effectiveness, and user satisfaction. The optimization of resource allocation is important in CC due to the dynamic and complex nature of cloud infrastructure. This study suggests a hybrid resource allocation procedure that combines Double Q-learning(DQ), a reinforcement learning (RL) approach, with meta-heuristic algorithms, including the Whale Optimization Algorithm (WOA), and it is compared with Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithms (GA) to optimize task allocation in CC settings. This combined system overcomes task allocation challenges such as usage of resources, energy consumption, and SLA compliance by integrating Q-learning's adaptive decision-making skills with meta-heuristic methods' global search efficiency. The proposed approach dynamically adapts optimum resource allocation guidelines while considering a variety of options for near-optimal task scheduling and resource assignment. The results of the experiment reveal that the hybrid strategy outperforms the PSO, ACO, and GA techniques in terms of makespan, reaction time, throughput, resource utilization of resources, cost, and rate of resource violation. This study provides an effective approach to handling complex resource allocation difficulties in a variety of dynamic cloud environments.**

Keywords*—Cloud computing; resource allocation; double Q-learning; meta-heuristic algorithms; Whale Optimization Algorithm (WOA); Particle Swarm Optimization (PSO); Ant Colony Optimization (ACO); Genetic Algorithms (GA)

I. Introduction

The rise of CC and services indicated the beginning of a new era of computing, giving numerous advantages, including expansion, flexibility, and affordability. Resource Allocation in CC is the process of distributing resources across different users and applications in the cloud. These resources include memory, computational power, storage, and network bandwidth [1]. The exponential rise of CC users has resulted in an extraordinary surge in demand for computing resources to perform cloud services. As a result, it is necessary to evaluate currently available resources such as virtual machines (VMs), CPU, RAM, and storage before allocating cloud services. The objective of resource allocation in the cloud is normally to assign resources to tasks (or virtual machines) in such a way that specific objectives are optimized, such as minimizing makespan, energy consumption, and maximizing resource utilization, throughput, and so on.

A. Cloud Computing

CC is a technological paradigm that enables customers to pay as they go for computing resources (e.g., servers, storage, databases, networking, software, and analytics) via the internet ("the cloud"). The design eliminates the need to own and manage physical data centers or servers, resulting in increased adaptability, scaling, and cost-efficiency. It is a model that provides on-demand access to a common pool of configurable computing assets (such as networks, storage, servers, applications, and services). These resources can be deployed and released with little management effort or intervention from the service provider. Cloud service models include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS offers virtualized computational assets such as virtual computers, storage, and networks. PaaS allows developers to construct, test, and launch apps without having to manage the structure that underlies them. Software as a service (SaaS) is delivered via the internet, generally through a web browser.

Resource allocation is a fundamental part of CC. It includes controlling and allocating the underlying virtual and physical resources in order to ensure efficient operation, fulfill user requests, and maintain service-level agreements. In CC, resource allocation focuses on optimizing resource distribution to improve efficiency and reduce costs while maintaining stability and scalability. It entails determining which resources to dedicate to specific activities, how much to give away, and for how long [2]. The basic concepts of resource allocation include the following.

Elasticity: The ability to gain resources in a flexible manner is seen as the fundamental reason for implementing a CC system. Elasticity primarily supports the ability to build and

shrink virtual resources dynamically in response to the needs of cloud users. Elastic resource allocation is critical for properly managing cloud resources based on customer demand and cloud service provider (CSP's) resources [3]. Elastic resources are programs and infrastructure that can be accessed on request when traffic or workloads increase. Cloud companies like AWS and Google Cloud use elastic resources as an enterprise model to bill consumers on demand, much like a utility bill. The cloud ecosystem is built on these supply and demand economic cycles. EC2 servers are an outstanding example of elastic resources. If a firm only needs two servers to host their website but experiences a holiday traffic increase, they may simply deploy more elastic resources by increasing the number of EC2 servers from two to four. Once the traffic decreases, they can reduce the server count to two.

Multi-tenancy: Multi-tenancy refers to having several tenants. This refers to a collection of users, resources, or services. The multi-tenant application was designed so that multiple users could access the same logic at the same time. A Software-as-a-Service (SaaS) provider can run numerous instances of the application and deliver web access to multiple customers, each with their own particular configuration. Such a model enables data isolation because each tenant's data is invisible to others.

QoS (Quality of Service): In CC, QoS is a metric used to assess the efficiency of a service, such as a network or cloud service. It is a method of evaluating the functionality of a service and ensuring that it fulfills the requirements. QoS measurements may include availability, reliability, and response time. The Cloud environment is a massive collection of virtually accessible resources that carry out millions of computing operations in real time for resource provisioning. Allocation and scheduling are two key components of providing for quality of service. This covers sophisticated modules including identifying work requirements, determining resource availability, making allocation decisions, and scheduling activities. In the current setting, it is challenging to manage cloud resources because service providers aim to deliver resources to consumers at productive cost and time[4].

Cost Efficiency: Inefficiencies in the cloud tend to manifest through over-provisioning and inefficient use of resources, which has the effect of increasing costs of operation. Common problems range from using too much CPU, storing too much data, and wasting on networking. If businesses overprovision their clouds, they pay for services not being utilized fully. Proper cloud cost optimization solutions are imperative in a bid to avoid such problems and enhance efficiency.

Workload Management: Strategic resource allocation, load balancing, and automation are required for effective cloud workload management to provide the highest achievable performance, efficiency, and cost savings in multiple cloud services and environments. Effective workload distribution in data centers enables business organizations to increase IT productivity, cut costs, and improve system effectiveness.

Dynamic resource allocation refers to the real-time assignment and reassignment of computing resources (e.g., CPU, memory, storage, and bandwidth) according to changing demands from users, software, or services in a cloud infrastructure. This method guarantees optimal resource utilization, cost efficiency, and responsiveness to dynamic workloads. Resource provisioning was done employing VM technology. It is argued that using a virtualized environment will reduce average job response times in addition to performing tasks on the basis of available resources. Consequently, users are allocated virtual machines based on work criteria. Efficient and dynamic utilization of cloud resources could help distribute the load and prevent occurrences such as system performance degradation. Meta-heuristics are techniques used to solve issues with a broad domain space that are difficult to research using conventional methods.

They are widely used in CC for resource allocation because of their capacity to handle complicated, multi-objective, and dynamic optimization problems. These algorithms have grown dynamically in previous decades. Scholars have been encouraged to use meta-heuristic algorithms to solve NP-hard issues. The benefit of employing these types of algorithms is that they can detect the best solution with less processing effort and fewer iterations than simple heuristic methods. The meta-heuristic method has some benefits, including simplicity, flexibility, source-independent solutions, and the ability not to become trapped in local optimum situations [5]. For effective resource allocation on the cloud, certain meta-heuristic algorithms such as ACO [6], PSO [7], Artificial Bee Colony (ABC)[8], Rock Hyrax Optimization (RHO)[9], and Bacterial Foraging Optimization (BFO)[10] were used. Resource distribution optimization methods are used to efficiently make allocations while satisfying constraints such as cost, power, and performance, and establishing balance among conflicting goals is very important in CC. The research goals are as follows:

- To design and test a hybrid resource allocation model incorporating Double Q-learning (DQ) and meta-heuristic algorithms to schedule tasks and allocate resources in dynamic CC environments efficiently.
- To compare the designed hybrid approach with other meta-heuristic algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithm (GA) in cloud computing environments.
- To attain optimal cloud performance metrics like makespan, response time, throughput, resource utilization, cost, and rate of resource violation.

The remaining part of this research is structured as follows. Section II discusses some existing works on resource allocation in cloud computing through reinforcement learning and meta-heuristics. Section III presents the proposed hybrid methodology by integrating Double Q-learning and the Whale Optimization Algorithm. Section IV discusses experimental findings and observations, and Section V concludes the work with key findings and future work directions.

II. Literature Review

In recent years, with the growing complexity of cloud resource scheduling requirements and the dynamic nature of cloud workloads, machine learning, deep learning,

reinforcement learning, and meta-heuristic optimization have played an increasing role in recent studies on cloud resource allocation. Their focus is on better use of resources, optimization of task scheduling, workload forecasting, load balancing, cost optimization, and service quality. Recently, there has been a lot of development going on in intelligent cloud resource management, which is summarized in the following studies.

Zheng et al. (2024) designed a prediction model by using XGBoost and LSTM to predict workload trends over time. The model is trained on historical data from an enormous cloud system with 1000 servers and more than 52 million data points. A dynamic resource scaling strategy is proposed with the use of the output of the prediction model and real-time system status inputs to make active allocation decisions [11]. Alizadeh et al. (2024) suggest an autonomous VM resource allocator based on the Clipped Double Deep Q-Learning (CDDQL) Algorithm and the meta-heuristic PSO. The tasks are prioritized through the use of the PSO algorithm, while CDDQL serves as the core of the autonomous system (Auto-CDDQL) in order to assign the proper VM resources to the tasks[12].

Zhang et al. (2024) present an action-constrained deep RL (DRL) technique for safely allocating computer resources and providing an optimal resource scheduling strategy. Task scheduling in dynamic cloud-based systems is a challenging task due to varying workload requirements, changing resource availability, and inconsistent task priorities. Traditional methods of optimization are generally inefficient in handling complex limitations [13]. To solve the above problems, Bajpai et al. 2024 introduce the Q-Whale algorithm (QWA) and the SARSA-Whale algorithm (SWA), a novel hybrid method that combines the WOA and RL methods [14]. Khani et al. review recent works on resource distribution in 5G cloud-based radio access networks (C-RAN) using DRL algorithms. It explores the potential of DRL to learn complex making decisions policies without intervention [15].

Gogula et al., 2024 try to formulate a task handling framework with decision-making capability to choose the best resource in real time for managing different and complicated applications on Virtual machines [16]. It applies a modified PSO in order to fulfill the user's deadline. Gongada et al., 2024 suggest a new task scheduling optimization framework that integrates Fruit Fly Optimization (FFO) and Convolutional Neural Networks (CNN). It highlights the importance of efficient resource allocation and management in CC to meet increasing demands for computing resources at reduced costs and enhanced overall system performance [17].

Aditi et al., 2025 compare a number of DL models, i.e., Recurrent Neural Networks, Residual Networks (ResNets), and CNN, to optimize the utilization of cloud resources and system performance [18]. Muniswamy et al.,2024 present a joint optimization of LB and resource allocation in the cloud with the help of an optimal container management method. It aims to enhance scheduling efficiency and be cost-saving through upgrading the requested remotely by users' container schedule. An enhanced backtracking search optimization (IBSO) method is applied to allocate resources between end-users/IoT devices and the cloud subject to service-level agreements [19]. Table I lists the recent related studies discovered in the literature for resource allocation in CC.

TABLE I. RELATED STUDIES IN THE LITERATURE FOR RESOURCE ALLOCATION IN CC

Author /Year	Methods	Data set	Evaluation Criteria	Limitations
Zheng et al., 2024	XGBoost and LSTM	Historical and real time data	Resource Utilization, Resource Violation Rate, Energy Efficiency	It works well on weel crafted features, not applicable for real time , computationally expensive
Alizadeh et al., 2024	CDDQL	c-hilo	Makes span, response time, task completion, resource utilization, and energy consumption rate	Computational complexity, takes a long time to converge, overfits to specific patterns
Zhang et al., 2024	DRL	Simulated data	Cost, delay, and security	High computational cost, long training time, scalability issues
Gogula et al., 2024	PSO	User-generated data	Resource utilization	Premature convergence, depends heavily on inertia weight and learning factors
Gongada et al., 2024	FFO and CNN	Historical data	Resource utilization and average response time	Only suitable for image processing
Aditi et al., 2025	CNN	MNIST dataset	Accuracy	Only suitable for image processing
Muniswamy et al.,2024	IBSO	Simulated data	Energy consumption, number of active servers, number of interruptions, total cost, runtime, and statistical measures.	Not suitable for real-time; requires tuning

Existing studies on cloud resource allocation have several limitations. Most of the approaches, such as XGBoost, LSTM, and DRL, are computationally costly, take a long time to train, and are not suitable for real-time applications. Approaches such as PSO are prone to premature convergence and are highly sensitive to parameter settings, whereas CNN-based models are primarily restricted to image processing tasks and are not suitable for application in different cloud environments. Besides, certain approaches have a tendency to overfit to particular data patterns and are not as agile as needed to handle dynamic and unpredictable workloads of cloud computing.

In spite of significant progress, there is an evident research gap in creating adaptive, scalable, and efficient resource allocation techniques for adaptive cloud environments. Existing works do not properly compromise between decision accuracy and global search ability, causing inefficient task scheduling and resource handling. Relatively few pieces of work have dealt with combining reinforcement learning methods such as Double Q-learning with meta-heuristic optimization techniques to combine local decision-making and global optimality. In addition, scarce research critically analyzes such hybrid approaches against multiple performance

measures like makespan, cost, response time, and SLA fulfillment.

III. METHODOLOGY

This section describes the technique utilized to achieve the research objectives. The hybrid technique utilizes the adaptive decision-making capabilities of Q-learning and the global exploration efficiency of meta-heuristic algorithms. The framework shown in Fig. 1 combines the information of the cloud workload with adaptive decision-making (based on Double Q-Learning) and global exploration and local exploitation (WOA). For comparison, PSO, ACO and GA are some alternative optimization techniques. The framework takes care of resource needs, checks out the candidate allocations, and generates optimal resource allocations taking into account utilization, performance, QoS, and SLA restrictions.

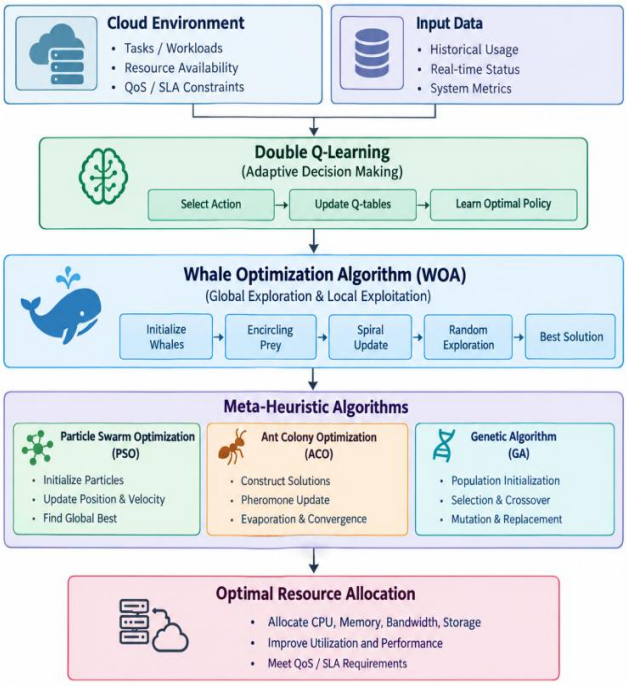


Fig. 1. Proposed hybrid double Q-learning and whale optimization algorithm framework.

A. Double Q-Learning

Q-learning is a RL method that teaches an agent how to perform optimal behaviors in a given environment in order to maximize cumulative rewards. It belongs to the value-based technique family and is especially beneficial for decision-making situations in which the agent communicates with the surrounding environment over time. The Bellman equation is used by the Q-learning algorithm to repeatedly update the Q-value for each state-action pair.

$$Q(s,a) \leftarrow q(s,a) + \alpha\{r + \max(Q(s',a') - Q(s,a)]) \quad (1)$$

where, Q(s,a): current estimate of the Q –value ,α , the learning rate , r –immediate reward after the action α in state s, γ-discount factor[20].

An improved version of the initial Q-learning algorithm, double Q-learning was derived to solve the problem of overestimation of the action value. In standard Q-learning, the value of an action is estimated using the maximum Q-value in the Q-table (or Q-function) for the subsequent state. As the action of maximum value is selected based on a noisy estimate instead of the actual value of the action, this could be a cause of overestimation. This bias of overestimation can cause inefficient policies or reduce learning speed. To overcome this bias, double Q-learning uses two separate Q-value estimators (Q1 and Q2). The steps in Double Q-learning are as follows.

Step 1: Two Q-value tables: Instead of maintaining a single Q-value table, Double Q-learning maintains two separate Q-value tables: $Q_1(s,a)$ and $Q_2(s,a)$ The two are updated independently.

Step 2: Action Selection: During learning, one of the Q-tables is used to select an action while learning.

Step 3: Update Process: While updating the Q-values, Double Q-learning employs the other Q-table to avoid the overestimation bias:

The action selected from the first Q-table (Q_1)is assessed according to the second Q-table (Q_2).

Particularly, the update of Q_1 and Q_2 is computed as follows:

$$Q_1(s,a) \leftarrow Q_1(s,a) + \alpha(r + \gamma Q_2(s', \arg\max_{a'} Q_1(s',a')) - Q_1(s,a)) \quad (2)$$

$$Q_2(s,a) \leftarrow Q_2(s,a) + \alpha(r + \gamma Q_1(s', \arg\max_{a'} Q_2(s',a')) - Q_2(s,a)) \quad (3)$$

This update employs the Q-table, which was not utilized to make the action choice, to calculate the next state's value and minimize overestimation of actions' values.

B. Whale Optimization Algorithm (WOA)

The WOA is a population-based optimization algorithm inspired by nature and based on humpback whale hunting behavior. The algorithm used belongs to the family of algorithms for swarm intelligence and was motivated by humpback whales' social behavior and hunting skills, especially their bubble-net feeding strategy. The WOA algorithmic Steps are:

- Randomly initialize a set of whales (solutions). Each whale carries an occupation in the solution space.
- Update Position: The algorithm updates the position of a whale by using three different strategies.
- Encircling Prey: Update the position of the whale using the distance between the current position of the whale and the best solution encountered so far (the prey).
- Spiral Update Position is inspired by the spiraling movement of whales while heading to the target. The update uses a spiral function in the equation.
- Random Exploration: During exploration, the algorithm uses random search to scan across the solution space.

- The above improvements make the algorithm mix global exploration (scanning widely across the solution space) with local exploitation (scanning narrowly around the current best so far).
- Objective Function Assessment: Each whale assesses the objective function (fitness function) so that it can identify how good the solution is. The fitness function is problem-specific.
- Update Best Solution: The algorithm holds the best solution obtained during the search and updates it.
- Termination: The algorithm updates the position of the whales at regular intervals and examines solutions until a stopping condition is met.

The flowchart for the hybrid Q-Whale approach is shown in Fig. 2.

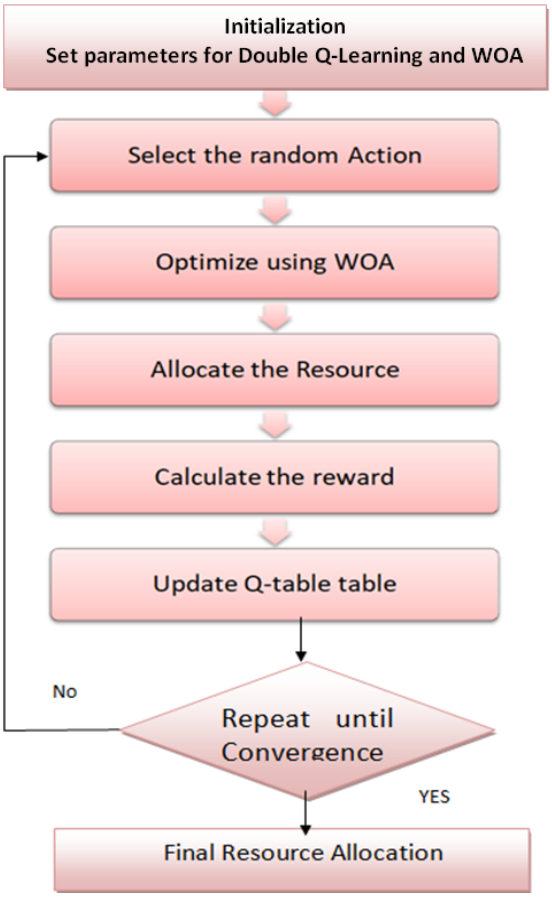


Fig. 2. Flowchart for the hybrid double Q-learning and WOA.

C. Particle Swarm Intelligence (PSO)

PSO is a school of fish or flocking-inspired algorithm that has been highly successful in addressing complicated optimization challenges, particularly in cloud computing resource allocation, communication networks, and energy consumption. In resource allocation, PSO can be utilized to allocate the resources (e.g., CPU capacity, memory, bandwidth, or storage) among the various tasks, users, or services in a way that is most cost-effective, efficient, and satisfies Quality of Service (QoS) needs.

PSO consists of a collection of candidate solutions known as particles, and each particle is associated with one of the alternatives in the solution space. All the particles move their position in the solution space based on their local information and information of adjacent particles. The key steps in PSO are:

- Problem Description: Particle position indicates the possibility of resource allocation. For instance, an allocation can be manifested as a vector of resource allocation over many tasks or users. The objective function relates to how usable or optimal a given allocation is, e.g., least cost, energy consumption, or makespan not surpassing intermediate resource limits.
- The initialization step: Particles are initialized randomly in a solution space. Each particle is an instance of a unique distribution of resources.
- The fitness function calculates the performance metrics (e.g., cost, QoS, energy usage) and requirements (e.g., SLA, resource availability) of a solution of each particle. For example, in CC, the fitness function can optimize idle resources, load balance, or reduce energy consumption with respect to quality of service constraints.
- Location and Velocity Update: The position and velocity of each particle are updated based on the PSO equations, considering its individual best and global best solution. The algorithm encourages good search in the solution space by a balance of exploration (search for new areas) and exploitation (local improvement of the current good solutions).
- Convergence: Particle positions are updated at intervals and converge to the global optimal solution, and make the fullest use of the resources in the long run. The algorithm stops when repetitions are at maximum or when improvement in solution stops [21].

D. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a nature-based optimization algorithm imitating ants' foraging behavior and, more significantly, how pheromone communication enables them to select the best path to food sources. ACO performs best on extremely large combinatorial optimization problems such as CC resource allocation, network optimization, scheduling jobs, and energy distribution. The general ACO method for resource allocation is as follows:

Initialization: The pheromone values are initialized on each edge (or decision variable) of the graph at the beginning. Ants are allocated to the graph in an effort to avoid experimenting with various schemes of resource allocation.

Solution Construction: One ant develops the solution by moving from a node (resource) to another, applying a probability decision rule on pheromone values and a heuristic function.

Pheromone Update: Once the solution is constructed, ants consider the quality of the solution with respect to an objective function (e.g., cost or performance). The concentration of pheromone on traveled paths is updated based on solution quality. Paths to better quality solutions have greater pheromone concentration and thus are more likely to be visited by following ants.

Evaporation: So that locally optimal solutions are avoided, the pheromone concentration decreases over time, and thus the impact of past routes is restricted.

Convergence: With more and more ants searching through different allocation paths, at some point the process converges to a phase where it uses an approximately optimal or optimal means of resource allocation.

ACO provides an efficient and universal solution to resource optimization in CC computer networks. Its capability of exploring diversified solutions and wide application in various situations renders it qualified for adaptive large-scale cloud systems with a dynamic nature. However, scalability, dynamic adaptability, and multi-objective optimization for practical use are all areas to be tackled [22].

E. Genetic Algorithms (GA)

Genetic Algorithms (GAs) are a powerful, nature-inspired optimization method based on genetics and natural selection. GAs have been used successfully to solve allocation problems in numerous fields, including cloud computing. Genetic algorithms are best suited to address complicated optimization problems, such as the resource allocation problem in cloud infrastructure, where the solution space is extremely huge, and objectives may be conflicting.

GAs operate on a population of possible solutions in which each solution is an individual member of the population. In cloud resource allocation, there can be various representations based on the given problem, which are: Chromosome Encoding: A GA individual or a chromosome might represent resource allocation task-specific. Each gene of the chromosome would be coded to some activity or resource, and resource allocations would be calculated based on the value of the gene (e.g., virtual machine to an activity, CPU or memory allocated, etc.). The standard Genetic Algorithm resource allocation process is the following:

Initialization: Random population of solutions (chromosomes) is created. Each solution contains one valid resource allocation.

Fitness Test: Each individual (resource allocation) is evaluated on the basis of a fitness function. The quality of the solution is defined by the fitness function. For example, it may penalize allocations that use more than available resources, excessive power consumption, or long runtimes.

Selection: Fittest individuals are selected on the basis of fitness degree. One may apply different selection strategies, for example:

- Roulette Wheel Selection: Probabilistically select individuals based on fitness.
- Tournament Selection: A method of choosing a pair at random from a set of people and selecting the winner.

Crossover (recombination) is an operation wherein two parent solutions are blended to produce offspring. In the case of resource allocation, this would be swapping portions of the chromosome (resource allocations) between the two individuals with the hope of creating new resource allocation solutions.

Mutation creates small random changes in offspring. It is done by assigning an asset at random to some other task or by assigning assets to replicate varying dynamics in the environment.

Replacement: The offspring are then inserted into the population, replacing the originals partially, usually based on their fitness.

Termination: The above process is run for a finite number of generations or until it achieves an optimal solution (i.e., population converges). Genetic algorithms are best suited to address resource allocation problems in cloud computing. They can search large spaces and optimize multi-objective problems with high levels, making them best suited to optimize resources in clouds. Scalability, variable environments, and convergence must be addressed prior to being implemented in actual clouds[23].

Algorithm: Hybrid Q-Whale Resource Allocation

Input: Cloud tasks, available resources, QoS/SLA constraints
Output: Optimized resource allocation
1. Initialize Q1, Q2 and the whale population.
2. Evaluate the initial whale solutions.
3. Identify the best whale solution.
4. Repeat until convergence:
   a. Observe the current cloud state.
   b. Select an action using Double Q-Learning.
   c. Execute the action and obtain the reward.
   d. Update Q1 or Q2 using the other Q-table.
   e. Update whale positions using WOA.
   f. Evaluate candidate allocations.
   g. Update the best solution.
   h. Check resource and QoS/SLA constraints.
5. Return the best resource allocation.

IV. RESULTS AND FINDINGS

This section discusses the tools and the parameters used to evaluate the resource allocation strategies. The proposed DQ-Whale resource allocation framework has been simulated and tested in the CloudSim simulation environment. A diverse cloud infrastructure was simulated using virtual machines with varying CPU sizes, memory amounts, and bandwidth and storage sizes. Dynamic workloads were generated by multiple cloud tasks that had different levels of computational requirements, priorities, and QoS constraints. The algorithms (DQ-Whale, PSO, ACO, and GA) were all implemented with the same workload and resources to facilitate a fair comparison. In simulation, task allocation and scheduling

decisions were continually analyzed depending on the makespan, throughput, response time, resource utilization, cost, and resource violation rate. The experiments were repeated with the same setup to evaluate the allocation efficiency and scheduling performance.

- Makespan: Total time needed to perform all tasks, and reducing makespan is a typical goal in resource allocation [24].
- Response time: The time it takes for a system to reply to a task request. Lower response times suggest more effective resource allocation [25].

$$Response\ time = completion\ time - task\ arrival\ time$$

- Throughput: Measures the number of jobs accomplished per unit of time. High throughput suggests efficient resource allocation [26].
- Resource Utilization: It tracks how effectively cloud resources (such as CPU, memory, and storage) are used. It represents the percentage of the resources accessible that are actively utilized. High utilization indicates efficient resource deployment but may result in overburden [27].

$$\text{Utilization rate} = \frac{Resource\ used}{Total\ Resource\ capacity} * 100\%$$

- Cost: The total cost taken for task allocation; lower values indicate cost effectiveness
- Resource violation rate: The percentage of time resources are overutilized or underutilized. A lower rate indicates better performance [28].

For more reliability in the experimental evaluation process, five different simulation runs of CloudSim were performed in each of the experiments with identical workload and resources of the five algorithms. All the same task characteristics, resources limitations and evaluation conditions were used for the DQ-Whale, PSO, ACO and GA to make a consistent comparison. The results are presented as mean ± standard deviation (n = 5 trials), and thus give information not only about the average performance in the trials, but also about the variation between the trials. Evaluated metrics: Makespan, throughput, response time, resource utilization, allocation cost, and resource violation rate. These results are therefore only observations of the algorithms based on a specific set of simulation conditions and should not be considered as statistically significant proof of the superiority of the algorithms.

Table II lists the performance measure values for the proposed Q-Whale, PSO, ACO, and GA algorithms. The makespan time for Q-Whale is 20 which is found to be low compared to all the other methods, indicating better performance in the hybrid Q-Whale algorithm. A value of 50 tasks/mts shows that a higher value in throughput indicates that the number of tasks completed per minute is higher in Q-Whale. The average time taken to respond to the task is low in the case of Q-whale. Q-Whale is cost-effective, and the resource violation rate shows only 2%, which is higher for Q-Whale compared to the contrasted methods PSO, ACO, and GA.

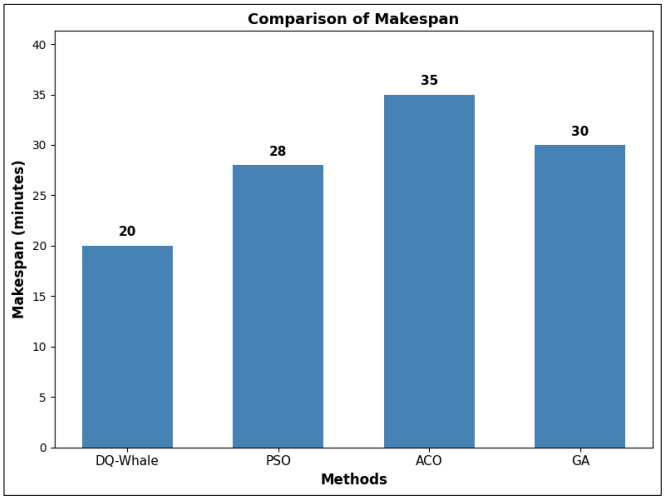


Fig. 3. Methods vs. makespan.

Fig. 3 shows a shorter makespan time of 20 minutes for the hybrid Q-Whale method and a maximum of 35 minutes for ACO. The nature of makespan time in the allocation of resources refers to how the overall completion time of a group of tasks (makespan) responds to or is influenced by various variables and methods in a resource allocation system. It is an important performance metric in CC and distributed systems since it has a direct impact on resource efficiency and system productivity.

TABLE II. PERFORMANCE MEASURES FOR RESOURCE ALLOCATION STRATEGIES

Methods/ Measures	Make span (mts)	Throughput (tasks/mts)	Response time (seconds/tasks)	Resource utilization (%)	Cost (units)	Resource violation Rate (%)
DQ-Whale (mts)	20	50	5	95	1000	2
PSO	28	45	8	88	1200	5
ACO	35	40	10	85	1500	7
GA	30	42	7	90	1300	4

Fig. 4 shows higher throughput for DQ-whale than for the PSO, ACO and GA methods. Throughput in resource allocation refers to how effectively a system can handle jobs over time, and it is an important performance indicator in CC, distributed systems, and task scheduling. Throughput influences a system's overall productivity and capacity to satisfy performance objectives.

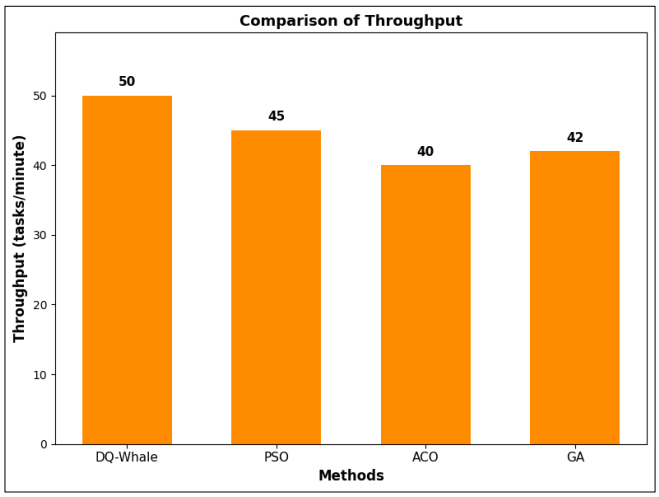


Fig. 4. Methods vs throughput.

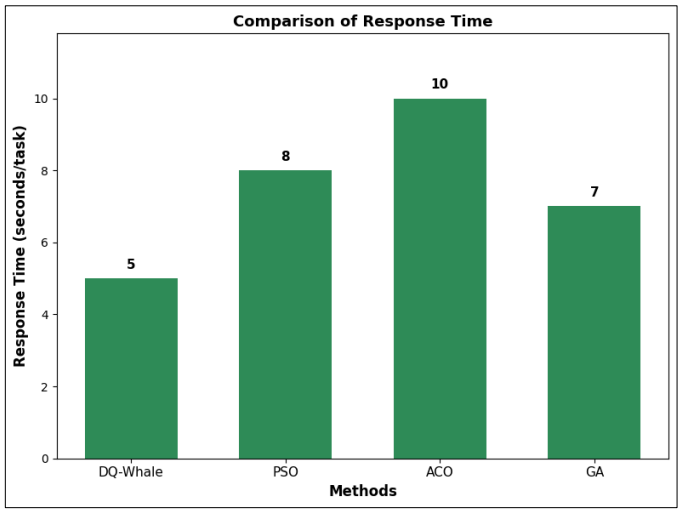


Fig. 5. Methods vs. response time.

Fig. 5 presents the response time for the proposed DQ-Whale method and other contrasted methods used for evaluation. The time taken to get a reply for tasks is low for the DQ-Whale method. The concept of reaction time in resource allocation relates to how soon a system can reply to and process a request or task once it has been received. It is an important performance indicator in computers, cloud services, and distributed systems since it impacts both user experience and system efficiency. Techniques that are adaptive to current network conditions, distribute load equitably, and properly prioritize jobs can assist in reducing reaction time.

As shown in Fig. 6, the resource utilization for DQ-Whale shows 95%, PSO as 88%, ACO as 85% and GA as 90% and is found to be high for hybrid DQ-Whale methods than the other methods. Resource utilization in resource allocation describes how efficiently and effectively the assets of a system (such as CPU, memory, bandwidth, and so on) are used to complete tasks. Proper resource use is critical for increasing performance, lowering expenses for operations, and ensuring that systems run efficiently.

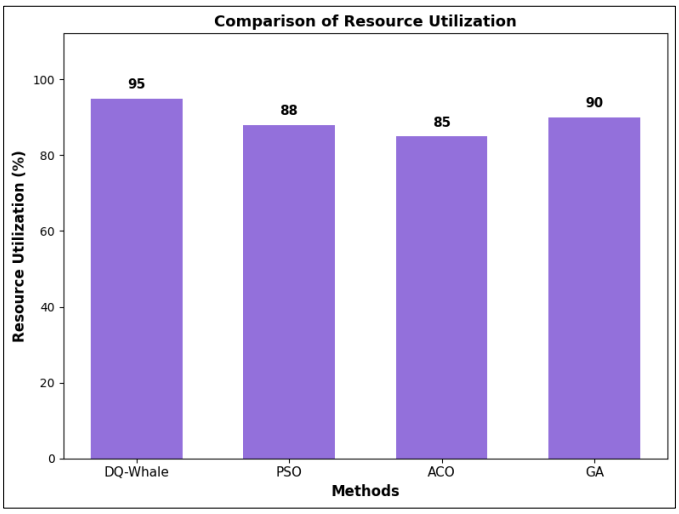


Fig. 6. Methods vs. resource utilization.

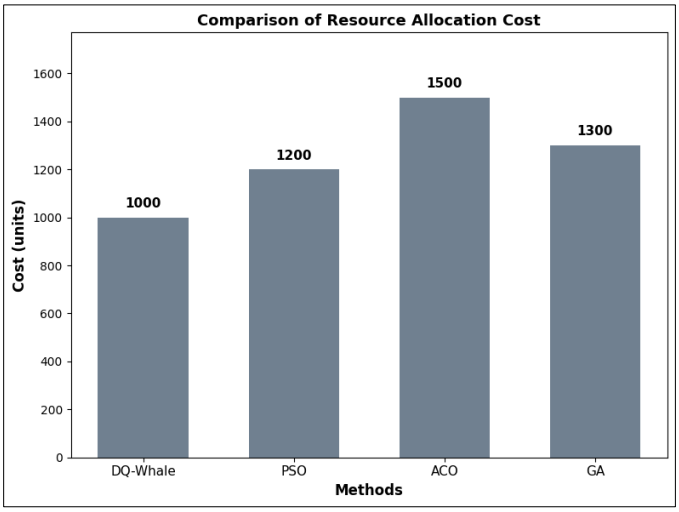


Fig. 7. Methods vs. cost.

Fig. 7 shows the decreased cost in units for proposed Q-whale method than the PSO, ACO and GA techniques. The cost in resource allocation relates to the numerous expenditures paid in order to properly use and manage computer resources inside a system. Cost concerns are critical in resource allocation because they impact decisions about infrastructure, optimizing performance, and overall system efficiency.

Fig. 8 shows a lower resource violation rate for DQ-Whale algorithm than for the PSO, ACO and GA strategies. In resource allocation, the resource violation rate refers to the frequency or possibility of resource consumption exceeding the allotted limitations or restrictions for a job or system. Resource violations can signal inefficiencies, future system overloads, or policy breaches; therefore, knowing this statistic is critical for improving resource allocation techniques. Q-Whale, PSO, ACO, and GA are examples of algorithms that aim to maximize throughput by efficiently distributing jobs to available resources. Their efficacy is determined by their ability to adapt to changing workloads, investigate the solution space, and strike a balance between exploration and exploitation.

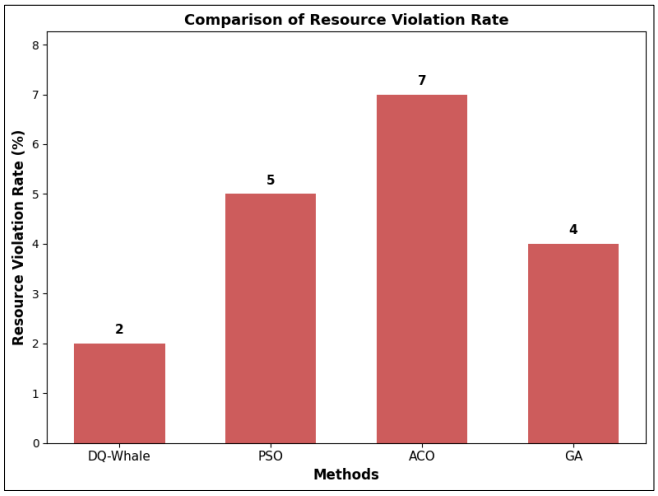


Fig. 8. Methods vs. resource violation rate.

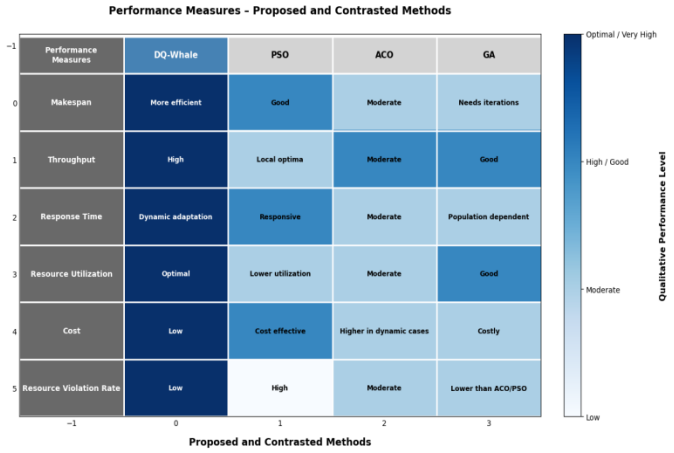


Fig. 9. Qualitative performance comparison of DQ-Whale, PSO, ACO, and GA across cloud resource allocation measures.

Fig. 9 is a comparison of DQ-Whale, PSO, ACO and GA in terms of six measures of cloud resource allocation. The makespan, throughput, latency, resource utilization, cost, and resource violation rate are good properties of DQ-Whale. It is adaptive in decision-making, which enables faster allocation and efficient utilization of resources with varying workloads. PSOs offer responsive allocation and can have local optima issues. ACO has moderate performance and convergence rate, especially in complex tasks. Good resource allocation is provided by GA, but it may need more iterations based on the complexity of the workload and population size. In general, the qualitative comparison shows the various features of the methods evaluated and the wide applicability of DQ-Whale for cloud resource management.

Table III points out that the main difference in methodology of DQ-Whale is the combination of Double Q-Learning with WOA, not the use of WOA. The reinforcement learning-WOA algorithms are also used as Q-Whale and SARSA-Whale, but with standard Q-Learning and SARSA, respectively. The innovation of DQ-Whale is that it employs two value estimators to diminish the potential overestimation in the process of action-value learning and keeps the WOA-based exploration and exploitation. The table also highlights that there is no numerical superiority of these methods over Q-Whale or SARSA-Whale for the experiments since the latter two methods were not tested with the same CloudSim configuration as described in the present experiments.

TABLE III. COMPARATIVE CHARACTERISTICS OF REINFORCEMENT-LEARNING–BASED HYBRID RESOURCE ALLOCATION APPROACHES

Performance/Methodological Aspect	DQ-Whale (Proposed)	Q-Whale	SARSA-Whale
Reinforcement learning strategy	Double Q-Learning	Q-Learning	SARSA
Value estimation	Two Q-tables	Single Q-table	Single action-value function
Action policy	Adaptive action selection using two value estimates	Q-value-based action selection	On-policy action selection
Overestimation control	Explicitly reduces Q-value overestimation	More susceptible to Q-value overestimation	Does not specifically employ Double Q-learning correction
Optimization mechanism	Whale Optimization Algorithm (WOA)	Whale Optimization Algorithm (WOA)	Whale Optimization Algorithm (WOA)
Exploration	Q-learning exploration combined with WOA exploration	Q-learning exploration combined with WOA	SARSA policy exploration combined with WOA
Exploitation	Double Q-value guidance with WOA exploitation	Q-value guidance with WOA exploitation	SARSA-based policy with WOA exploitation
Adaptation to changing states	High, through double-estimator learning	Adaptive through Q-value updates	Adaptive through on-policy updates
Allocation decision mechanism	State–action learning with reduced value-estimation bias	State–action learning	State–action–next-action learning
Main methodological characteristic	Combines bias-reduced reinforcement learning with population-based optimization	Combines conventional Q-learning with WOA	Combines on-policy SARSA learning with WOA
Expected computational consideration	Additional Q-table maintenance	Lower Q-learning memory requirement	Additional policy-transition evaluation
Direct comparability with present study	Evaluated under the proposed CloudSim configuration	Requires identical reimplementation and experimental conditions	Requires identical reimplementation and experimental conditions

There are still several issues in the current methods of allocation of cloud resources that can be improved in the future. A variety of studies have been conducted on algorithms in simulated and relatively controlled workloads, and this may not accurately reflect workloads in a highly dynamic cloud environment. Other approaches can have a tendency to

converge prematurely, suffer from local optima, have a higher computational load, and lack scalability with the number of tasks and resources. Moreover, few works have focused on multi-objective optimization that considers the trade-offs among the cost, energy consumption, latency, resource utilization, and QoS at the same time. Following researchers can consider adaptive hybrid algorithms with real world cloud workloads, online learning, multi objective optimization and distributed decision making in the future. Performance in edge-cloud and heterogeneous environments, as well as in case of workload uncertainty, failures of resources, and varying service level requirements, should be further studied.

V. CONCLUSION

The proposed Double Q-Learning–Whale Optimization (DQ-Whale) framework is an adaptive framework to allocate cloud resources using the decision-making process of the Double Q-Learning algorithm along with the exploration and exploitation ability of the Whale Optimization Algorithm (WOA). The experimental outcomes shown with CloudSim demonstrate good performance when compared with PSO, ACO, and GA for the simulation conditions with respect to the metrics, such as makespan, throughput, response time, resource utilization, cost and resource violation rate. But there are some drawbacks to this approach, DQ-Whale. Two Q-tables and a number of whale solutions increase the computational and memory cost; its performance might be dependent on the WOA parameters like the number of the whale population, the number of iterations and exploration–exploitation coefficients. Furthermore, the current assessment is based on a defined workload scale and simulated cloud environment, and thus the scalability, i.e., the ability to scale up to much larger workloads and significantly more heterogeneous infrastructures, has yet to be proven. In future, the numbers of tasks and virtual machines will be gradually increased to study scalability, parameter sensitivity, and ablation studies for the WOA, and computational overhead will be measured. Further experiments will investigate robustness against dynamic workloads and failures of resources, as well as changing QoS/SLA requirements. Repeated trials will also be performed together with the appropriate statistical significance tests for extensions of the multi-objective incorporating the energy consumption, cost, latency, and QoS.

REFERENCES

[1] Mohammad, A., & Abbas, Y. (2024). Key Challenges of Cloud Computing Resource Allocation in Small and Medium Enterprises. *Digital*, *4*(2), 372-388. https://doi.org/10.3390/digital4020018

[2] Alam, Shahzeb, Muhammad Atif Ramzan, Muhammad Zubair, Ubaid Ullah, Rab Nawaz, and Rahmat Ullah. "AI-driven resource allocation in cloud computing: a systematic review revealing critical sustainability and evaluation gaps." *Computing* 108, no. 5 (2026): 67.

[3] Thapa, Mitali, Kamal Kant Hiran, Simran Ajit Kumar Mohanty, Deepak Parashar, and Ankur Goyal. "Elastic Cloud Scaling for Small Models: Dynamic Resource Allocation as an Alternative to Training Large LLMs." *IEEE Access* (2026).

[4] Vali, Ali Akbar, Sadoon Azizi, Mohammad Shojafar, and Rajkumar Buyya. "Energy-efficient resource management in microservices-based fog and edge computing: State-of-the-art and future directions." *ACM Computing Surveys* 58, no. 12 (2026): 1-42.

[5] S. Singhal, N. Gupta, P. Berwal, Q. N. Naveed, A. Lasisi and A. W. Wodajo, "Energy Efficient Resource Allocation in Cloud Environment Using Metaheuristic Algorithm," in IEEE Access, vol. 11, pp. 126135-126146, 2023, doi: 10.1109/ACCESS.2023.3330434.

[6] Lilhore, Umesh Kumar, Sarita Simaiya, KBV Brahma Rao, VVR Maheswara Rao, Yogesh Kumar Sharma, Roobaea Alroobaea, Hamed Alsufyani, Majed Alsafyani, and MD Monish Khan. "Hybrid DRL-enhanced ACO-WWO for efficient resource allocation and load-balancing in cloud computing." *International Journal of Computational Intelligence Systems* 18, no. 1 (2025): 148.

[7] Nabi, Said, Masroor Ahmad, Muhammad Ibrahim, and Habib Hamam. "AdPSO: adaptive PSO-based task scheduling approach for cloud computing." *Sensors* 22, no. 3 (2022): 920.

[8] Kruekaew, Boonhatai, and Warangkhana Kimpan. "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning." *Ieee Access* 10 (2022): 17803-17818.

[9] Singhal, Saurabh, Ashish Sharma, Pawan Kumar Verma, Mohit Kumar, Sahil Verma, Maninder Kaur, Joel JPC Rodrigues, Ruba Abu Khurma, and Maribel García-Arenas. "Energy efficient load balancing algorithm for cloud computing using rock hyrax optimization." *IEEE access* 12 (2024): 48737-48749.

[10] Hu, Yanjuan, Leiting Pan, Wenjun Lv, and Zhanli Wang. "Optimal selection of cloud manufacturing resources based on bacteria foraging optimization." *International Journal of Computer Integrated Manufacturing* 37, no. 1-2 (2024): 165-182.

[11] Zheng H, Xu K, Zhang M, Tan H, Li H. Efficient resource allocation in cloud computing environments using AI-driven predictive analytics. Applied and Computational Engineering. 2024 Sep 10;82:17-23.

[12] Alizadeh Javaheri SD, Ghaemi R, Monshizadeh Naeen H. An autonomous architecture based on reinforcement deep neural network for resource allocation in cloud computing. Computing. 2024 Feb;106(2):371-403.

[13] Zhang H, Wang J, Zhang H, Bu C. Security computing resource allocation based on deep reinforcement learning in serverless multi-cloud edge computing. Future Generation Computer Systems. 2024 Feb 1;151:152-61.

[14] Bajpai G, Singh P, Agarwal AK. Performance Enhancement of Cloud Resource Management by Using Optimized Reinforcement Learning Approaches. Journal of Computational Analysis and Applications (JoCAAA). 2024 May 20;33(05):977-91.

[15] Khani M, Jamali S, Sohrabi MK, Sadr MM, Ghaffari A. Resource allocation in 5G cloud-RAN using deep reinforcement learning algorithms: A review. Transactions on Emerging Telecommunications Technologies. 2024 Jan;35(1):e4929.

[16] Gogula S, Sridhar P, Arvind S, Sethy A, Ragavendiran SP, Balasubramani P, Gupta KG. Optimized resource allocation in cloud computing for enhanced performance with modified particle swarm optimization. InMATEC Web of Conferences 2024 (Vol. 392, p. 01140). EDP Sciences

[17] Gongada TN, Desale GB, Ghodake SP, Sridharan K, Rao VS, El-Ebiary YA. Optimizing Resource Allocation in Cloud Environments using Fruit Fly Optimization and Convolutional Neural Networks. International Journal of Advanced Computer Science & Applications. 2024 May 1;15(5).

[18] Aditi, Prasad VK, Gerogiannis VC, Kanavos A, Dansana D, Acharya B. Utilizing convolutional neural networks for resource allocation bottleneck analysis in cloud ecosystems. Cluster Computing. 2025 Feb;28(1):22.

[19] Muniswamy S, Vignesh R. Joint optimization of load balancing and resource allocation in cloud environment using optimal container management strategy. Concurrency and Computation: Practice and Experience. 2024 May 30;36(12):e8035

[20] Uma, J., Vivekanandan, P., & Shankar, S. (2022). Optimized intellectual resource scheduling using deep reinforcement Q-learning in cloud computing. Transactions on Emerging Telecommunications Technologies, 33(5), e4463.

[21] Bahrepour, D., Evaznia, N., & Khodabakhshi, T. (2024). A New Resource Allocation Method Based on PSO in Cloud Computing. International Journal of Web Research, 7(2), 13-21.

[22] Yin, C., Fang, Q., Li, H., Peng, Y., Xu, X., & Tang, D. (2024). An optimized resource scheduling algorithm based on GA and ACO algorithm in fog computing. The Journal of Supercomputing, 80(3), 4248-4285.

[23] Behera, I., & Sobhanayak, S. (2024). Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach. Journal of Parallel and Distributed Computing, 183, 104766.

[24] K. Karim, F., Ghorashi, S., Alkhalaf, S., HA Hamza, S., Ben Ishak, A., & Abdel-Khalek, S. (2024). Optimizing makespan and resource utilization in cloud computing environment via evolutionary scheduling approach. PloS one, 19(11), e0311814.

[25] Alizadeh Javaheri, S. D., Ghaemi, R., & Monshizadeh Naeen, H. (2024). An autonomous architecture based on reinforcement deep neural network for resource allocation in cloud computing. Computing, 106(2), 371-403.

[26] Jeyaraman, J., Bayani, S. V., & Malaiyappan, J. N. A. (2024). Optimizing Resource Allocation in Cloud Computing Using Machine Learning. European Journal of Technology, 8(3), 12-22.

[27] Gong Y, Huang J, Liu B, Xu J, Wu B, Zhang Y. Dynamic resource allocation for virtual machine migration optimization using machine learning. arXiv preprint arXiv:2403.13619. 2024 Mar 20.

[28] Mukherjee, A., De, D., & Buyya, R. (2024). Cloud Computing Resource Management. In Resource Management in Distributed Systems (pp. 17-37). Singapore: Springer Nature Singapore.

Generative AI for Data-Informed Educational Decision-Making: A Systematic Review and Framework

Husna Hafiza Razami[1], Abtar Darshan Singh[2], Roslina Ibrahim[3],
Ali Najeeb[4], Helio Augusto da Costa Xavier Mauquei[5]
Digital Learning Hub, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia[1, 2]
Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia[3]
Qasim Ibrahim School of Business, Villa College, QI Campus, Rahdhebai Magu, Male, Maldives[4]
Management Department of the Faculty of Economics, Universidade Nacional Timor Lorosa'e,
Rua Avenida Cidade de Lisboa, Dili, Timor-Leste[5]

***Abstract*—Generative artificial intelligence (GenAI) is creating new possibilities for data-informed decision-making in higher education, yet how it translates educational data into actionable decisions for different stakeholders remains undertheorised. This study reports a PRISMA-guided systematic review of 38 peer-reviewed empirical studies addressing three questions: what educational decisions GenAI supports in higher education (RQ1); how educational data and GenAI capabilities enable those decisions (RQ2); and what framework explains GenAI-driven educational decision-making (RQ3). Across the 38 studies, eight decision categories are identified inductively and organised into micro (learner-facing), meso (instructor and program), and macro (institutional) levels. Five enabling GenAI capabilities (natural language generation, behavioural and semantic analysis, multimodal processing, synthetic data generation, and conversational analytics) are mapped to each category. Synthesising these findings, this study proposes the GenAI-Driven Educational Decision-Making (GDEDM) framework: a four-layer model connecting educational data sources, GenAI capabilities, educational decision levels, and governance. The evidence reveals important cross-level patterns, divergences, and research gaps, exposing a critical governance gap between rapidly advancing GenAI capabilities and lagging institutional safeguards. The GDEDM framework offers researchers, practitioners, and policymakers a coherent, evidence-grounded basis for responsible GenAI deployment in higher education.**

Keywords*—Educational decision-making; generative artificial intelligence; higher education; data-informed decisions; large language models; systematic review

I. Introduction

Higher education institutions have invested in data-driven approaches to support a wide range of operational goals, including identifying at-risk students, personalising academic pathways, evaluating teaching effectiveness, and informing institutional resource allocation [1]. Yet translating educational data into timely, actionable decisions for the right stakeholders has remained a persistent challenge.

The emergence of generative artificial intelligence (GenAI), which encompasses large language models (LLMs) such as GPT-4, open-source models including Llama and ERNIE Bot, retrieval-augmented generation (RAG) systems, and generative adversarial networks (GANs), has significantly expanded what is technically possible in this space[2], [3], [4]. Unlike traditional machine learning approaches that predict or classify based on structured input, GenAI systems can generate natural language, synthesise insights from complex data, and support multimodal analysis [4]. These capabilities position GenAI as a bridge between educational data and decision-making, enabling systems to move beyond producing analytical outputs toward supporting educational decisions for students, instructors, and administrators [5], [6].

Early data-informed decision support in higher education relied on machine learning models for academic outcome prediction, including identifying at-risk students, forecasting grades, and modelling dropout risk [7]. These systems demonstrated the value of multi-feature models integrating behavioural, academic, and socioeconomic variables, but their outputs, such as probability scores and classification results, required considerable human expertise to translate into practical action.

The emergence of LLMs changed what AI-generated educational outputs look like: where predictive systems produce numbers, GenAI systems produce text, recommendations, and dialogues that function as decision artifacts in themselves [3]. Standards-based interoperable architectures combining multi-agent systems, xAPI trace collection, and LTI delivery have further shown how such GenAI decision outputs can be embedded within institutional learning management ecosystems at scale [2]. Yet despite this progress, the field lacks a comprehensive, decision-centred understanding of what GenAI supports across the full spectrum of higher education stakeholders and organisational levels.

Most existing reviews organise findings by model type, analytical technique, or isolated function (e.g., focusing on predictive tasks [8], dashboard applications [9], human-centered design [10], or explainability [11]) rather than by the educational decisions GenAI enables or the governance conditions required to make those decisions responsible. The present study addresses this gap through a systematic review of 38 empirical studies guided by three research questions:

- RQ1: What educational decisions are supported by GenAI in higher education?
- RQ2: How do educational data and GenAI capabilities support these decisions?
- RQ3: What framework can explain GenAI-driven educational decision-making in higher education?

The study makes three original contributions. First, a decision-centric synthesis of 38 empirical studies organised around eight inductively derived decision categories that provide an outcome-focused perspective on how GenAI supports educational decision-making. Second, a three-level decision taxonomy (micro, meso, and macro) grounded entirely in the empirical evidence. Third, the GDEDM framework: a four-layer, governance-inclusive model linking educational data sources, GenAI capabilities, educational decision levels, and responsible AI governance to support both scholarly understanding and institutional deployment.

II. Methodology

A. Search Strategy and Inclusion Criteria

Electronic searches were completed on March 6, 2026 across Scopus, Web of Science, and IEEE Xplore. The search string combined GenAI terms ("generative artificial intelligence" OR "Generative AI" OR "large language model*" OR LLM OR ChatGPT OR GPT) with educational data and decision terms ("learning analytics" OR "educational decision" OR "educational data mining" OR "student analytics") and higher education qualifiers ("higher education" OR universit* OR college OR "tertiary education" OR undergraduate* OR postgraduate*). Backward and forward citation snowballing supplemented database retrieval.

Studies were included if they: (IC1) employed a GenAI technique, such as a large language model or generative adversarial network; (IC2) applied it within a learning analytics (LA) workflow; (IC3) were conducted in a higher education context; (IC4) were published in a peer-reviewed journal or conference proceeding; and (IC5) reported original empirical research. A study was considered to involve educational data or LA workflow if it used learner-related data for analysis, modelling, or interpretation to inform understanding or support decision-making in an educational context [12], [13].

Following deduplication, 244 records were screened by title and abstract, of which 35 progressed to full-text review and 24 met the eligibility criteria. Citation snowballing identified 14 additional eligible studies, yielding a final corpus of 38 studies. The main exclusion reasons were absence of GenAI-enhanced LA (EC2, n = 121), non-empirical design (EC6, n = 51), absent GenAI application (EC1, n = 34), non-higher education setting (EC3, n = 7), and non-peer-reviewed source (EC4, n = 3). Some records met multiple criteria. Two reviewers independently screened the records, resolving disagreements through discussion and consensus, particularly regarding the presence of a learning analytics workflow and empirical evidence. The complete selection process is presented in Fig. 1.

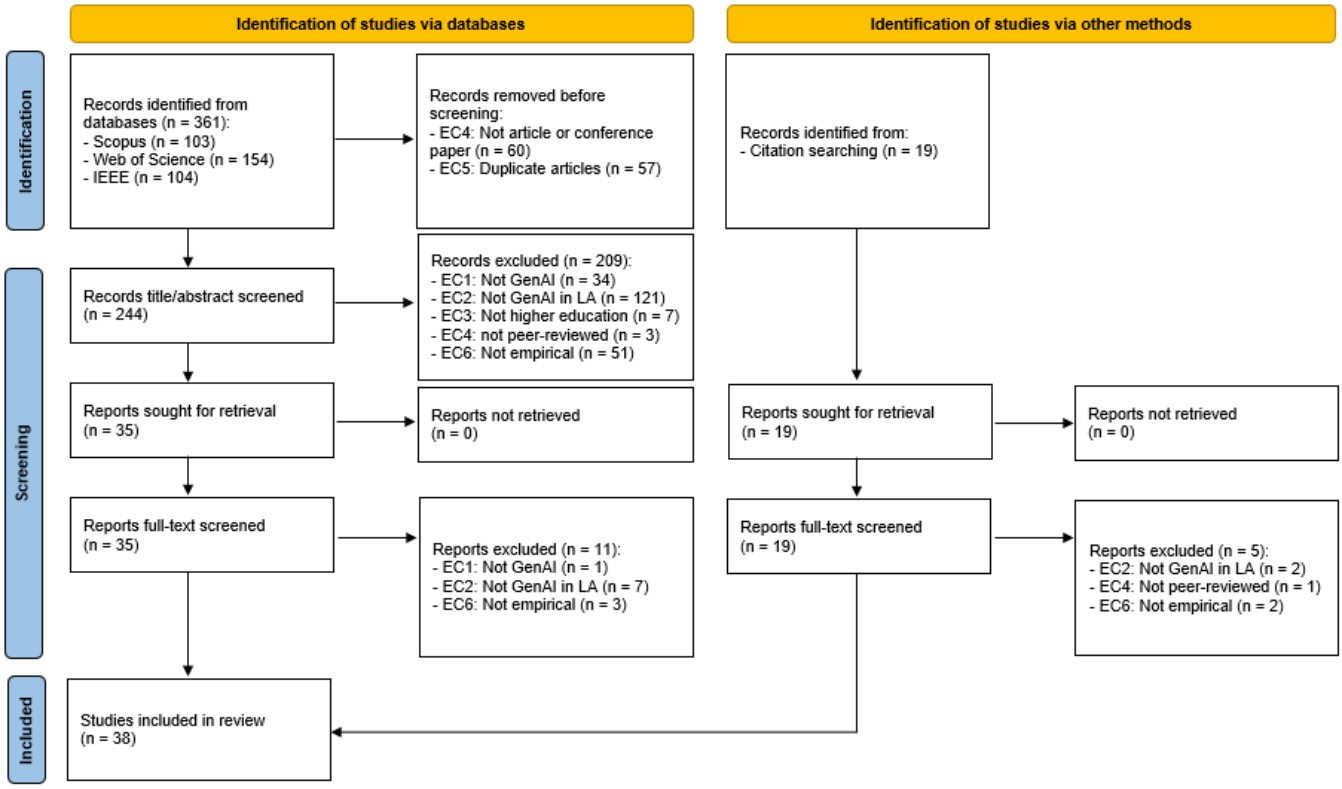


Fig. 1. PRISMA 2020 flow diagram of the article selection process.

B. Data Extraction and Analytical Approach

A structured extraction form captured the GenAI models, educational data sources, stakeholders served, educational decision types, and the GenAI capabilities enabling those decisions. The decision categories were derived inductively. For each study, the decision output was identified and grouped with functionally similar outputs across other studies. Categories were iteratively refined until each was conceptually distinct.

The eight resulting categories were then organised into three decision levels based on stakeholder scope and systemic consequence. Similarly, GenAI capabilities were identified by examining the primary role each GenAI technique performed within the learning analytics workflow. Functionally similar capabilities were grouped and iteratively refined into five overarching categories representing how GenAI enabled educational decision-making across the reviewed studies. Data extraction was performed by the first author and verified by a co-author, with discrepancies resolved through discussion.

Study quality was assessed using the Quality Assessment with Diverse Studies (QuADS) tool [14], a 13-item appraisal instrument suited to the heterogeneous mix of system evaluations, case studies, and user studies in the corpus. The first author appraised each included study across all 13 items, with total scores ranging from 24 to 38 out of 39, indicating generally adequate methodological quality. Quality appraisal informed the depth of synthesis and confidence in specific findings.

III. Results

A. Educational Decisions Supported by GenAI (RQ1)

The synthesis of the 38 studies identified eight educational decision categories supported by GenAI in higher education. These categories were organised into three decision levels: micro (learner-facing), meso (instructor- and program-level), and macro (institutional). Table I maps the decision categories to the GenAI technologies reported in the reviewed studies. Studies employing multiple technologies appear in more than one technology column. This section introduces the decision categories and associated technology patterns, while the subsequent RQ2 section examines how educational data and GenAI capabilities enable these decisions.

1) Micro-level decision categories: Four categories operate at the individual learner level. Feedback and Assessment Support is the most frequently studied category in the corpus (n = 8) and one of the broadest in technology coverage, supported by GPT-family models, open-source LLMs, unspecified LLMs, and RAG systems. This category focuses on assessment-related decisions through personalised feedback, formative assessment, rubric-based evaluation, and other feedback mechanisms that support learner improvement. The prominence of Feedback and Assessment Support may reflect the growing demand for scalable assessment support and personalised feedback in higher education.

TABLE I. Educational Decision Categories and Supporting GenAI Technologies

Decision Level	Decision Category	GPT-4/4o (ChatGPT)	Open-source LLMs (Llama, DeepSeek, ERNIE)	Unspecified LLMs	RAG Systems	Multimodal LLMs	GANs	Other Generative Models (TVAE, Gaussian Copula, DALL-E)	SQL / NL Query Agents	n
Micro	Feedback and Assessment Support	[15], [16], [17], [18], [19], [20]	[15]	[21]	[22]					8
	Learning Pathway Personalisation	[23], [24], [25], [26]			[22]	[23], [25]				5
	Academic Guidance and Advising	[6], [27], [28], [29]						[29]		4
Meso	Instructional Design Support	[25], [30], [31], [32]	[30], [31], [33]			[25]				5
	Collaborative Learning Analytics	[23], [34], [35]	[34], [36]	[37]		[38]				6
	Analytics Interpretation and Sensemaking	[5], [18], [39], [40]			[39], [41]	[39], [41]			[5], [40]	5
Macro	Early Warning and Intervention	[26], [42], [43], [44]	[45]				[46], [47]	[44]		7
	Strategic Institutional Planning	[5], [27], [29], [40], [48]	[45], [48]					[29]	[5], [40], [49]	7
	Total unique studies per technology	**27**	**8**	**2**	**3**	**5**	**2**	**3**	**3**	

2) Learning pathway personalisation (n = 5) supports decisions about adapting learning experiences, content sequencing, and instructional pathways to individual learner needs. Unlike Feedback and Assessment Support, which evaluates completed learning activities and provides formative feedback, this category focuses on adapting future learning experiences to individual learner needs. It exhibits considerable technological diversity, incorporating GPT-family models, RAG systems, and multimodal LLMs. Among the reviewed studies, it is the only micro-level category incorporating multimodal LLMs, suggesting growing interest in using diverse learner inputs to support personalised learning experiences.

Academic Guidance and Advising (n=4) supports decisions concerning course selection, programme planning, and academic progression. Whereas Learning Pathway Personalisation focuses on adaptation within learning environments, advising applications address broader academic choices that shape learners' educational trajectories. The category relies predominantly on GPT-family models, with only one study reporting the use of another generative approach (DALL-E) [29]. Its relatively narrow technology profile suggests that a limited range of GenAI approaches has been explored for advising applications in the reviewed studies. Collectively, the three micro-level categories demonstrate the central role of large language models in supporting learner-facing educational decisions, while highlighting distinct forms of decision support ranging from feedback to personalisation and advising.

3) Meso-level decision categories: Three categories serve instructors and program teams. Collaborative Learning Analytics (n=6) supports decisions related to monitoring, evaluating, and facilitating collaborative learning activities. It is the largest meso-level category and exhibits substantial technological diversity, combining GPT-family models, open-source LLMs, multimodal LLMs, and unspecified LLM-based agents across the reviewed studies. This technological diversity may reflect the complexity of collaborative learning, which often involves interactions captured through multiple modalities, including text, video, and behavioural traces [38].[38]

Instructional Design Support (n=5) supports curriculum improvement, course redesign, and instructional planning. Whereas Collaborative Learning Analytics focuses on understanding learner interactions, this category focuses on modifying the learning environment through evidence-informed instructional decisions. GPT-family models, open-source LLMs, and multimodal LLMs are the primary technologies reported in this category.

Analytics Interpretation and Sensemaking (n=5) focuses on supporting stakeholders in interpreting and acting on learning analytics outputs. Rather than generating new instructional recommendations, its primary role is to help stakeholders interpret and act upon existing analytical findings. Among the reviewed studies, it is the only meso-level category incorporating Structured Query Language (SQL) and natural-language query agents and one of only three categories employing Retrieval-Augmented Generation (RAG) systems. This category highlights the role of GenAI in bridging the gap between learning analytics outputs and their practical use in educational decision-making.

4) Macro-level decision categories: Two categories operate at the institutional level. Early Warning and Intervention (n = 7) supports decisions related to identifying at-risk students and initiating timely interventions through institution-wide monitoring and intervention strategies. It incorporates a distinctive combination of GPT-family models, graph neural networks (GNNs), GAN-based approaches, and other generative models, reflecting the use of complementary AI techniques to support institution-wide risk identification and intervention.

Strategic Institutional Planning (n = 7) supports institution-wide strategic and operational decisions related to student success, enrolment management, resource allocation, and organisational performance. Whereas Early Warning and Intervention focuses on identifying students requiring timely support, Strategic Institutional Planning focuses on informing broader institutional planning and management decisions. Its technology profile is characterised by GPT-family models, SQL-based systems, natural-language query agents, and other generative models, highlighting the role of conversational access to institutional data in supporting evidence-informed organisational decision-making.

Together, these macro-level categories illustrate how GenAI supports complementary forms of institutional decision-making: Early Warning and Intervention focuses on identifying and responding to student risk, whereas Strategic Institutional Planning supports broader organisational planning and management. Compared with the learner- and programme-focused categories, macro-level applications rely more heavily on institution-wide educational data and support decisions with broader organisational scope and consequences.

5) Technology distribution and structural gaps: GPT-family models appear across all eight decision categories and are employed across 27 of 38 studies, indicating their position as the dominant and most widely adopted technology in the current GenAI decision-support landscape. Open-source large language models (LLMs) (8 studies) appear across several educational decision categories, often serving as privacy-conscious or locally deployable alternatives to proprietary GPT-family models. Retrieval-Augmented Generation (RAG) systems and multimodal large language models (MLLMs) (3 and 5 studies, respectively) occupy more specialised roles, with RAG supporting knowledge-grounded applications and MLLMs concentrated in contexts involving multimodal educational data. Other generative models, including TVAE, Gaussian Copula, and DALL-E, appear in three studies, supporting synthetic data generation and complementary generative tasks. GAN-based models are identified within Early Warning and Intervention, where they enable privacy-preserving synthetic data generation for predictive learning analytics. SQL-based systems and natural-language query agents appear only in Analytics Interpretation and Sensemaking and Strategic Institutional Planning.

One notable structural gap in Table I is the absence of RAG-based systems within macro-level decision categories. Although RAG technologies are represented in several micro- and meso-level applications, none of the reviewed macro-level studies identified their use. This pattern suggests a potential area for future research, particularly given the growing importance of transparency, traceability, and evidence-informed decision-making in higher education institutions. Whether this absence reflects implementation challenges, research priorities, or the relative novelty of RAG technologies remains unclear from the current evidence base. The next section builds on these findings

by examining how educational data and GenAI capabilities support the identified educational decisions.

B. Data and GenAI Capabilities Enabling Educational Decisions (RQ2)

Table II summarises the educational data sources, stakeholders, and GenAI capabilities associated with each decision category across the 38 reviewed studies.

1) Mechanisms at the micro-level: Individual learner data form the foundation of all three micro-level decision categories. Across Feedback and Assessment Support, Learning Pathway Personalisation, and Academic Guidance and Advising, studies draw on assessment responses, LMS interaction logs, reflection texts, discussion forum contributions, student engagement data, and navigation histories to identify learner characteristics, engagement patterns, and learning needs. Natural language generation (NLG), often combined with conversational support mechanisms, translates these insights into personalised feedback, recommendations, adaptive guidance, and other learner-facing outputs that support educational decision-making.

TABLE II. DECISION-LEVEL MAPPING: STAKEHOLDERS, DATA SOURCES, GENAI CAPABILITIES, AND STUDIES

Decision Level	Decision Category	Stakeholder(s)	Educational Data Source	GenAI Capability	Studies
Micro (Learner-Facing)	Feedback and Assessment Support	Students, Instructors	Student handwritten notes, writing assignments, open-ended responses, discussion forum posts, online activity data, LMS interaction logs, feedback trace logs	NLG for personalised and scalable feedback; RAG-grounded formative assessment generation; LLM rubric interpretation and tag-based recommendation; dialogue-based scaffolding of analytics feedback	[15], [16], [17], [18], [19], [20], [21], [22]
	Learning Pathway Personalisation	Students, Instructors	Multimodal data (speech, facial expression, text), discussion forum posts, student engagement and behavioural data, large-scale MOOC datasets, real-time navigation and activity logs	Adaptive scaffolding generation from real-time trace data; multimodal chain-of-thought pathway planning; RAG-guided learning objective alignment; NLP-based engagement analysis to support adaptive personalisation	[22], [23], [24], [25], [26]
	Academic Guidance and Advising	Students, Academic advisors	Student course enrolment data, advising survey responses, advisor evaluation records, focus group transcripts, AI-assisted learning interaction data	Profile-aware NLG course recommendation; LLM rule-based course matching; GPT-based major selection advising with advisor comparison; qualitative thematic analysis for student perspective understanding	[6], [27], [28], [29]
Meso (instructor/ Program)	Instructional Design Support	Instructors, Curriculum designers	Multimodal classroom data (speech, facial expression, text), question bank datasets, student reflective writing, student interaction and behavioural trace logs, course data	Multimodal LLM analysis for iterative course redesign; LLM-based automated learning objective labeling; reflection analysis for instructor insights; ERNIE Bot-based human-AI collaboration analysis; LLM co-creation of instructional materials	[25], [30], [31], [32], [33]
	Collaborative Learning Analytics	Instructors, Researchers, Administrators	Discussion forum posts, asynchronous peer discussion datasets, mixed reality interaction logs, student reflection texts, video recordings of collaborative tasks, multimodal data (eye-tracking, interaction logs, dialogue)	LLM-based cognitive presence detection and summarisation; automated SSRL phase coding; process mining of knowledge construction sequences; grounded prompt engineering for deductive coding (GROPROE); multimodal LLM posture annotation; help-seeking process analysis	[23], [34], [35], [36], [37], [38]
	Analytics Interpretation and Sensemaking	Students, Instructors, Administrators	Online learning activity logs and peer review data, descriptive LA task datasets, LA dashboard interaction datasets, student reflection and GenAI assistant interaction logs	GPT-4o + RAG conversational agents for dashboard interpretation; NL-based dashboard and visualisation generation; ChatGPT code interpreter for descriptive analytics support; multimodal GenAI + RAG scaffolding for visual analytics comprehension; dialogue-based scaffolding of analytics feedback	[5], [18], [39], [40], [41]
Macro (Institutional)	Early Warning and Intervention	Instructors, Advisors, Academic support staff	Campus card transaction and exam score data, student engagement and behavioural data, lecture slides and assessment records, ChatGPT interaction logs, e-book interaction logs including reading time and page interactions	LLM + GNN graph embedding for multi-source behavioural analysis and risk detection; GPT-4 + NLP for engagement-driven predictive insights; LLM-driven ontology extraction and knowledge graph construction for performance prediction; ChatGPT-based feature extraction from conversation logs; LLM agent for e-book interaction factor analysis	[26], [42], [43], [44], [45]
	Strategic Institutional Planning	Administrators, Institutional researchers	Retention data, GPA trends and graduation rate records, campus card and exam score data, student survey responses, focus group transcripts, online learning datasets, descriptive LA task datasets, AI-assisted learning interaction data	Automated SQL query generation for institutional data access; LLM + GNN multi-source report generation; LLM-based sentiment analysis and summarisation of student surveys; GPT-4 + Claude qualitative thematic coding; GPT-4o NL dashboard generation and visualisation; ChatGPT code interpreter for non-expert analytics support; GPT-Turbo generative learning insights	[5], [27], [29], [40], [45], [48], [49]

Retrieval-Augmented Generation (RAG) strengthens several micro-level applications by grounding outputs in domain-specific educational content, including course rubrics, learning objectives, and disciplinary knowledge, thereby improving factual consistency and contextual relevance [22], [39]. Studies employing RAG generally emphasise grounding mechanisms as a means of improving the reliability and contextual relevance of generated outputs. A complementary quality-assurance pattern is also evident. Studies without RAG frequently incorporate instructor review between GenAI output generation and learner delivery (Xavier et al., 2025; Lekan & Pardos, 2025), whereas RAG-enabled systems are more often implemented with higher levels of automation. This pattern suggests that both RAG and human oversight may serve as complementary mechanisms for enhancing output quality and reliability at the micro level.

Despite relying on similar GenAI technologies, the micro-level decision categories differ in the learner data they use and the decisions they support. Feedback and Assessment Support focuses on assessment-related decisions through personalised feedback, formative assessment, rubric interpretation, and other feedback mechanisms that support learner improvement. In contrast, Learning Pathway Personalisation uses behavioural, engagement, and multimodal learner data to adapt future learning experiences, content sequencing, and instructional pathways to individual learner needs. Academic Guidance and Advising combines learner records, advising information, and student perspectives to support course selection, programme planning, and academic progression. These differences reflect distinct educational objectives and decision contexts despite their reliance on similar underlying GenAI capabilities. Treating them as a single analytical category would therefore obscure important differences in the learner data, decision processes, and evaluation approaches associated with each category.

2) Mechanisms at the meso-level: The analytical challenge at the meso level shifts from individual personalisation to cohort-level pattern detection. Collaborative Learning Analytics draws on one of the widest ranges of educational data in the corpus, including discussion forum posts, video recordings, group activity logs, student reflections, and multimodal interaction data. Across the reviewed studies, multimodal LLM annotation and process mining are used to extract meaningful pedagogical signals from these collaborative interactions [37], [38], [50]. A nuanced distinction exists between behavioural coding (classifying what interaction types occurred) and process mining (mapping temporal sequences and dynamics of collaboration): both are present in the evidence base, but few studies combine them. The limited integration of these two analytical modes suggests an opportunity for richer and more actionable collaborative decision support.[37], [38], [50]

Instructional Design Support uses aggregated cohort data to generate curriculum and instructional recommendations, typically through a pattern analysis stage that informs subsequent NLG [30], [31]. GenAI in this category identifies which learning objectives, topics, or instructional sequences are associated with persistent underperformance, then generates specific redesign suggestions for instructors.

Analytics Interpretation and Sensemaking relies on a highly diverse capability set, combining natural-language querying, RAG-augmented explanation, and dialogue-based scaffolding to support stakeholders with varying levels of technical expertise, from non-technical administrators querying databases [40] to students interpreting their own performance patterns [18]. Unlike the other meso-level categories, the primary contribution of this category lies not in generating new analytical insights but in enabling stakeholders to understand and act upon existing learning analytics through conversational and context-aware interactions.

Together, the three meso-level categories illustrate complementary mechanisms through which GenAI supports instructor- and programme-level decision-making. Collaborative Learning Analytics focuses on extracting pedagogical insights from complex interaction data, Instructional Design Support translates cohort-level patterns into curriculum improvements, and Analytics Interpretation and Sensemaking makes analytical outputs more accessible to stakeholders through natural-language interaction and contextual explanation. These differences demonstrate that GenAI supports meso-level decisions not only by generating new insights but also by improving stakeholders' ability to interpret and act on existing learning analytics.

3) Mechanisms at the macro-level: At the macro level, educational decisions are informed by institution-wide educational data, including student information systems, retention records, survey responses, campus card transactions, and longitudinal behavioural data. These broader and more integrated data sources enable GenAI capabilities that support institution-wide risk identification, strategic planning, and organisational decision-making.

Early Warning and Intervention combines multiple educational data sources to identify students at risk of academic failure or disengagement. The reviewed studies employ complementary GenAI technologies, including graph neural networks (GNNs), large language models (LLMs), and GAN-based synthetic data generation. GNNs are used to model the relational and temporal structure of student behavioural data, while LLMs translate analytical outputs into natural-language risk summaries and intervention recommendations [26], [42], [45]. GAN-based approaches generate privacy-preserving synthetic educational data that support the development, evaluation, and validation of predictive learning analytics models when authentic educational data are limited [46], [47]. Together, these complementary capabilities strengthen institution-wide risk identification and intervention.

Strategic Institutional Planning draws on institutional datasets, including graduation records, retention data, survey responses, and administrative databases, to support strategic and operational decisions. Natural-language querying and SQL-based interfaces improve access to institutional analytics by enabling administrators to interrogate complex data through conversational interaction, thereby supporting evidence-informed planning and decision-making [5], [40], [49].

4) Cross-level patterns, contradictions, and research gaps: Across the reviewed studies, three cross-level patterns emerge. First, learner-generated educational data, including LMS interaction logs, assessment records, and behavioural traces, form the primary foundation for GenAI-supported educational decisions across all three levels, although multimodal and institutional data become increasingly important at the meso and macro levels. This pattern reflects the growing maturity of educational data infrastructures supporting GenAI applications in higher education [2].

Second, human oversight varies across decision levels. At the micro level, studies report both augmentation approaches, where instructors review GenAI-generated outputs before learner delivery [6], [16], and automation approaches, where outputs are delivered with minimal human intervention [17], [20]. In contrast, macro-level applications consistently position institutional stakeholders as the primary decision-makers, with GenAI supporting rather than replacing institutional judgment [5], [40], [49].

Third, GenAI capability selection appears to depend on the educational task. Studies comparing proprietary GPT-family and open-source LLMs for learner-facing natural language generation generally report stronger performance for GPT-family models in the reviewed studies for feedback and advising tasks [6], [15]. In contrast, studies employing open-source LLMs for behavioural coding and analytical tasks report comparable performance while highlighting advantages in cost, privacy, and local deployment [34], [35], [36]. At the macro level, complementary techniques such as GNNs and GANs extend GenAI capabilities by supporting relational modelling and privacy-preserving synthetic data generation for predictive learning analytics [42], [46], [47]

Among the research gaps identified across the reviewed studies, limited evidence on the long-term effectiveness of GenAI-supported educational decisions and the relatively limited attention to responsible AI governance emerged as particularly important. Only one study [17] evaluated outcomes across multiple semesters, while fairness, transparency, and accountability were seldom addressed despite the increasing use of GenAI in educational decision-making. These cross-level patterns and research gaps informed the development of the GDEDM framework presented in the following section.

C. The GDEDM Framework (RQ3)

The findings from RQ1 and RQ2 reveal a consistent structural logic across the evidence: educational data provide the foundation for GenAI capabilities, which enable educational decisions operating within responsible AI governance. The GDEDM (Generative AI-Driven Educational Decision-Making) framework formalises this logic as four interconnected layers, illustrated in Fig. 2. Unlike frameworks that describe analytical workflows or system architectures, GDEDM is organised around the conceptual relationships between educational data, GenAI capabilities, educational decisions, and responsible AI governance, making it applicable as both a scholarly explanatory model and a practical reference for institutional GenAI adoption.

1) Layer 1 – Educational data sources: Layer 1 comprises the educational data that underpin all GenAI-supported educational decisions. The framework distinguishes two complementary data sources: learner-generated data, representing learning activities, assessment, and behavioural traces, and institutional data, comprising academic and administrative records that support institution-wide decision-making. The quality and representativeness of these data determine which educational decisions can be supported and how reliably they can be informed. Richer multimodal data, including classroom recordings, eye-tracking, and posture traces, expand decision support to collaborative and immersive learning contexts [37], [38], [50] while introducing additional privacy obligations and processing complexity. Consequently, data quality is fundamental to trustworthy decision-making.[37], [38], [50]

2) Layer 2 — GenAI capabilities: Layer 2 comprises five complementary GenAI capabilities: Natural Language Generation, Behavioural and Semantic Analysis, Multimodal Processing, Conversational Analytics, and Synthetic Data Generation. Natural Language Generation produces personalised feedback, recommendations, and other educational outputs; Behavioural and Semantic Analysis extracts meaningful patterns and insights from learner interactions and educational data; Multimodal Processing integrates and interprets heterogeneous educational data, including text, speech, video, and behavioural traces; Conversational Analytics enables natural-language interaction with learning analytics and institutional data, supporting query generation, dashboard visualisation, and contextual interpretation; and Synthetic Data Generation creates privacy-preserving datasets for developing, evaluating, and validating AI-supported educational decision systems when authentic educational data are limited.

Retrieval-Augmented Generation (RAG) functions as a cross-cutting quality mechanism within Layer 2 by grounding LLM outputs in verified educational content, thereby improving factual consistency and trustworthiness [22], [39]. Among the reviewed studies, RAG was not applied in macro-level decision support, representing an opportunity to enhance the transparency and reliability of institutional GenAI applications.

3) Layer 3 — Educational decisions: Layer 3 organises the eight evidence-derived educational decision categories into three decision levels. The micro level comprises learner-facing decisions (Feedback and Assessment Support, Learning Pathway Personalisation, and Academic Guidance and Advising); the meso level supports instructor- and programme-level decisions (Instructional Design Support, Collaborative Learning Analytics, and Analytics Interpretation and Sensemaking); and the macro level supports institution-wide decisions (Early Warning and Intervention and Strategic Institutional Planning).

Together, these levels represent a progression from individual learner support to organisational decision-making, with increasing stakeholder responsibility and governance requirements. This progression provides a practical roadmap for institutional GenAI adoption, enabling lower-risk learner-facing applications before extending deployment to institution-wide decision support requiring more comprehensive governance.

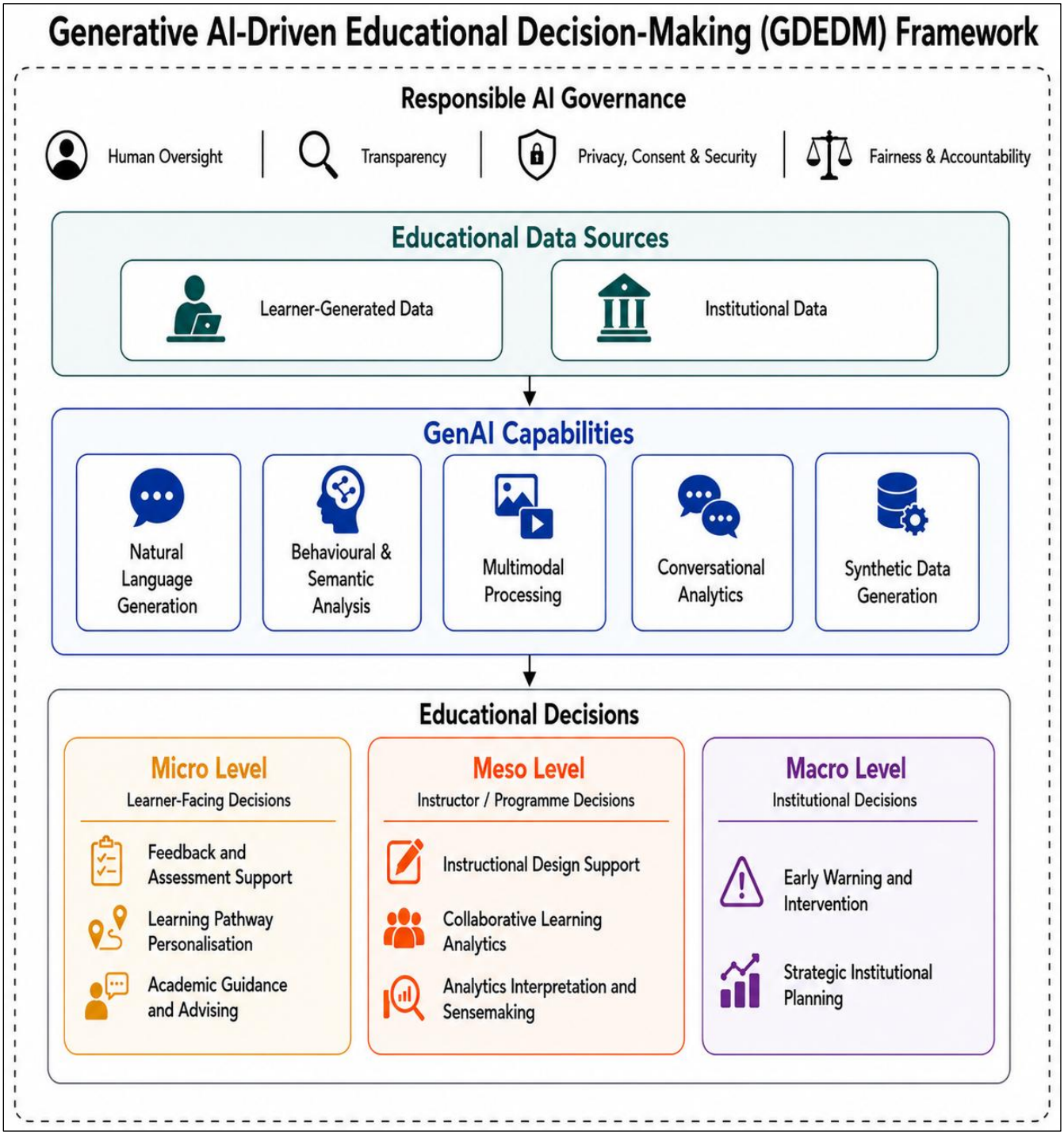


Fig. 2. The GDEDM framework.

4) Layer 4 — Responsible AI governance: Governance is represented as an overarching layer surrounding the framework, ensuring that responsible AI principles apply across educational data, GenAI capabilities, and educational decisions. The framework highlights four governance principles: Human Oversight, Transparency, Privacy, Consent and Security, and Fairness and Accountability. Human Oversight ensures meaningful human authority over consequential educational decisions. Transparency supports explainable GenAI outputs. Privacy, Consent and Security govern the responsible

management of educational data, while Fairness and Accountability require institutions to monitor bias and maintain clear responsibility for GenAI-supported decisions. Together, these principles embed governance across all layers of the framework rather than treating it as a separate implementation stage.

The review also highlights several governance priorities for institutional GenAI adoption. None of the reviewed studies systematically evaluated whether GenAI-supported decisions produced differential outcomes across student demographic groups despite well-established concerns regarding algorithmic bias [3], [7]. Consequently, the GDEDM framework emphasises fairness and accountability through equity auditing, particularly for macro-level decisions affecting institutional planning, resource allocation, and student success. Responsible deployment also requires transparency, informed consent, data minimisation, human oversight, and clearly defined accountability. Together, these governance principles support trustworthy GenAI-enabled educational decision-making.

IV. Discussion

The GDEDM framework and its evidential foundations offer three broader insights into the adoption of GenAI for educational decision-making in higher education.

First, organising the literature around educational decisions rather than GenAI technologies reveals research priorities that are less apparent in technology-centred reviews. The eight educational decision categories show that evidence is strongest for Feedback and Assessment Support (n = 8), whereas Academic Guidance and Advising (n = 4) and Instructional Design Support (n = 5) remain comparatively underexplored despite their importance for improving student success and teaching quality. These findings suggest that future research should extend beyond feedback generation to investigate how GenAI can support a broader range of educational decisions across learners, instructors, and institutions.

Second, the review identifies a governance gap between rapidly advancing GenAI capabilities and institutional safeguards. GenAI systems are increasingly being deployed to support decisions at the micro, meso, and macro levels, yet relatively few studies explicitly address governance mechanisms such as transparency, human oversight, privacy, or fairness. The GDEDM framework therefore positions responsible AI governance as a cross-cutting layer, demonstrating that governance should accompany educational data, GenAI capabilities, and educational decisions rather than being considered only after deployment.

Third, the review highlights explainability as a critical requirement for trustworthy educational decision-making. At the micro level, GenAI frequently delivers personalised feedback, recommendations, and guidance directly to learners. However, students and educators may be unable to judge whether these outputs are accurate, appropriate, or free from bias. Developing explainable GenAI systems that provide understandable justifications for educational recommendations therefore represents an important direction for future research and practical implementation.

These findings should, however, be considered in light of several limitations. The search was restricted to three databases and English-language publications, potentially introducing coverage and publication bias. In addition, the literature search was completed in March 2026; therefore, studies published thereafter were not included. Although grounded in 38 empirical studies, the GDEDM framework has not yet been independently validated through expert assessment or real-world institutional application.

V. Conclusion

This study presented a PRISMA-guided systematic review of 38 empirical studies examining how GenAI supports educational decision-making in higher education. For RQ1, eight educational decision categories were identified and organised into micro-, meso-, and macro-level decisions. Feedback and Assessment Support (n = 8), Early Warning and Intervention (n = 7), and Strategic Institutional Planning (n = 7) were the most frequently represented categories, whereas Academic Guidance and Advising (n = 4) remained comparatively underexplored.

For RQ2, five GenAI capability categories were identified, demonstrating how natural language generation, behavioural and semantic analysis, multimodal processing, conversational analytics, and synthetic data generation enable different forms of educational decision-making across stakeholders and institutional contexts. For RQ3, this study proposed the GenAI-Driven Educational Decision-Making (GDEDM) framework, a four-layer model connecting educational data sources, GenAI capabilities, educational decisions, and responsible AI governance. The framework provides both a conceptual explanation of GenAI-supported educational decision-making and a basis for future research and evaluation of responsible institutional adoption.

The review identifies three priorities for future research. First, longitudinal and multi-institutional studies are needed to evaluate the long-term effectiveness of GenAI-supported educational decisions. Second, systematic evaluation of fairness and accountability, including equity auditing, is required to support trustworthy institutional deployment. Third, explainable GenAI approaches should be developed to improve transparency and stakeholder trust across learner-, instructor-, and institution-facing applications. Overall, the GDEDM framework provides an evidence-based model that integrates educational data, GenAI capabilities, educational decisions, and responsible AI governance to support responsible GenAI adoption in higher education, thereby contributing to the advancement of inclusive and high-quality higher education in support of Sustainable Development Goal 4 (SDG 4: Quality Education).

References

[1] G. Siemens, “Learning Analytics: The Emergence of a Discipline,” American Behavioral Scientist, vol. 57, no. 10, pp. 1380–1400, 2013, doi: 10.1177/0002764213498851.

[2] A. M. Khaddar, Y. Said, A. Dehbi, and T. Chafiq, “An Interoperable Multi-Agent Architecture for Personalized Smart Learning Using Generative AI and Learning Analytics,” IJACSA) International Journal of Advanced Computer Science and Applications, vol. 17, no. 4, 2026, doi: 10.14569/IJACSA.2026.0170449.

[3] M. Á. Rodríguez-Ortiz, P. C. Santana-Mancilla, and L. E. Anido-Rifón, "Machine Learning and Generative AI in Learning Analytics for Higher Education: A Systematic Review of Models, Trends, and Challenges," Applied Sciences (Switzerland), vol. 15, no. 15, Aug. 2025, doi: 10.3390/app15158679.

[4] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," Learn. Individ. Differ., vol. 103, pp. 1–9, 2023, doi: 10.1016/j.lindif.2023.102274.

[5] Z. Wang, W. Lin, and X. Hu, "Self-service Teacher-facing Learning Analytics Dashboard with Large Language Models," in 15th International Conference on Learning Analytics and Knowledge, LAK 2025, Association for Computing Machinery, Inc, Mar. 2025, pp. 824–830. doi: 10.1145/3706468.3706491.

[6] K. Lekan and Z. A. Pardos, "AI-Augmented Advising: A Comparative Study of GPT-4 and Advisor-Based Major Recommendations," Journal of Learning Analytics, vol. 12, no. 1, pp. 110–128, Mar. 2025, doi: 10.18608/jla.2025.8593.

[7] R. Abdrakhmanov et al., "Development of a Framework for Predicting Students' Academic Performance in STEM Education using Machine Learning Methods," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 15, no. 1, 2024, doi: 10.14569/IJACSA.2024.0150105.

[8] N. Sghir, A. Adadi, and M. Lahmer, "Recent advances in Predictive Learning Analytics: A decade systematic review (2012–2022)," Educ. Inf. Technol. (Dordr)., vol. 28, no. 7, pp. 8299–8333, Jul. 2023, doi: 10.1007/s10639-022-11536-0.

[9] L. Cabral, R. Pinto, and G. Gonçalves, "AI-powered learning analytics dashboards: a systematic review of applications, techniques, and research gaps," Dec. 01, 2025, Discover. doi: 10.1007/s44217-025-00964-y.

[10] P. Topali, A. Ortega-Arranz, M. J. Rodríguez-Triana, E. Er, M. Khalil, and G. Akçapınar, "Designing human-centered learning analytics and artificial intelligence in education solutions: a systematic literature review," Behaviour and Information Technology, vol. 44, no. 5, pp. 1071–1098, 2025, doi: 10.1080/0144929X.2024.2345295.

[11] S. Gunasekara and M. Saarela, "Systematic Literature Review on Explainable Learning Analytics and Educational Data Mining," IEEE Access, vol. 13, pp. 214387–214408, 2025, doi: 10.1109/ACCESS.2025.3643645.

[12] D. Clow, "The learning analytics cycle: closing the loop effectively," in ACM International Conference Proceeding Series, New York, NY, USA, 2012, pp. 134–138. doi: 10.1145/2330601.2330636.

[13] L. Yan, R. Martinez-Maldonado, and D. Gasevic, "Generative Artificial Intelligence in Learning Analytics: Contextualising Opportunities and Challenges through the Learning Analytics Cycle," in ACM International Conference Proceeding Series, Association for Computing Machinery, Mar. 2024, pp. 101–111. doi: 10.1145/3636555.3636856.

[14] R. Harrison, B. Jones, P. Gardener, and R. Lawton, "Quality assessment with diverse studies (QuADS): an appraisal tool for methodological and reporting quality in systematic reviews of mixed- or multi-method studies," BMC Health Serv. Res., vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12913-021-06122-y.

[15] W. Dai et al., "Assessing the Proficiency of Large Language Models in Automatic Feedback Generation: An Evaluation Study Title Page," Computers and Education: Artificial Intelligence, vol. 7, 2024, doi: 10.1016/j.caeai.2024.100299.

[16] C. Xavier et al., "Empowering Instructors With AI: Evaluating the Impact of an AI-Driven Feedback Tool in Learning Analytics," IEEE Transactions on Learning Technologies, vol. 18, pp. 498–512, 2025, doi: 10.1109/TLT.2025.3562379.

[17] M. Yilmaz, A. Gauthier, and M. Cukurova, "Supporting online learners' regulation skills with the help of learning analytics and generative artificial intelligence," Educational Technology Research and Development, 2026, doi: 10.1007/s11423-026-10595-1.

[18] Y. Uzun, A. Gauthier, and M. Cukurova, "What Students Ask, How a Generative AI Assistant Responds: Exploring Higher Education Students' Dialogues on Learning Analytics Feedback," in Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference, New York, NY, USA: ACM, Apr. 2026, pp. 42–52. doi: 10.1145/3785022.3785028.

[19] Y. Y. Zhou, C. Tang, and A. Shimada, "A Novel Approach: Enhancing Data Extraction from Student Handwritten Notes Using Multi-Task U-net and GPT-4," in 2024 7th International Symposium on Autonomous Systems, ISAS 2024, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ISAS61044.2024.10552516.

[20] F. J. Y. Jin et al., "Students' Perceptions of Generative AI–Powered Learning Analytics in the Feedback Process: A Feedback Literacy Perspective," Journal of Learning Analytics, vol. 12, no. 1, pp. 152–168, Mar. 2025, doi: 10.18608/jla.2025.8609.

[21] N. Volkmann, "EduBot Unleashed - Elevating Digital Competence in Online Collaborative Learning," in 2024 21st International Conference on Information Technology Based Higher Education and Training, ITHET 2024, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ITHET61869.2024.10837628.

[22] S. Pozdniakov et al., "From knowledge gaps to learning opportunities: Leveraging student questions and dual use of generative AI to support student learning at scale," Computers and Education: Artificial Intelligence, vol. 10, Jun. 2026, doi: 10.1016/j.caeai.2025.100509.

[23] Q. Chen and K. Zhou, "Large Language Model-Based Multimodal Learning Analytics and Intelligent Path Planning," in 2025 6th International Conference on Information Science and Education (ICISE-IE), IEEE, Dec. 2025, pp. 415–419. doi: 10.1109/ICISE-IE68873.2025.11379719.

[24] T. Li et al., "Turning Real-Time Analytics into Adaptive Scaffolds for Self-Regulated Learning Using Generative Artificial Intelligence," in 15th International Conference on Learning Analytics and Knowledge, LAK 2025, Association for Computing Machinery, Inc, Mar. 2025, pp. 667–679. doi: 10.1145/3706468.3706559.

[25] Y. Fang, X. Chen, X. Fang, T. Xia, T. He, and X. Cao, "A Unified Multimodal Learning Analytics Framework for Course Design with LLMs," in 2025 International Conference on Artificial Intelligence and Education, ICAIE 2025, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 258–262. doi: 10.1109/ICAIE64856.2025.11158598.

[26] R. Sajja, Y. Sermet, D. Cwiertny, and I. Demir, "Integrating AI and Learning Analytics for Data-Driven Pedagogical Decisions and Personalized Interventions in Education," Technology, Knowledge and Learning, 2025, doi: 10.1007/s10758-025-09897-9.

[27] A. Bakharia, A. Shibani, L.-A. Lim, and S. B. Shum, "From Transcripts to Themes: A Trustworthy Workflow for Qualitative Analysis Using Large Language Models Analysis Using Large Language Models," in LAK 2025 Workshops, 15th International Conference on Learning Analytics and Knowledge(LAK 2025), Dublin, Ireland, 2025.

[28] M. A. Z. Khan, A. Polyzou, and N. Bennamane, "How Can We Use LLMs for EDM Tasks? The Case of Course Recommendation," in CEUR Workshop Proceedings, Atlanta, Georgia, USA, Jan. 2024.

[29] L. Abrusci, K. Dabaghi, and F. Sciarrone, "Learning Analytics Through Generative AI: An Exploratory Study," in Lecture Notes in Networks and Systems, Springer Science and Business Media Deutschland GmbH, 2026, pp. 141–152. doi: 10.1007/978-3-032-15743-0_12.

[30] N. Liu, S. Sonkar, D. Basu Mallick, R. Baraniuk, and Z. Chen, "Atomic Learning Objectives and LLMs Labeling: A High-Resolution Approach for Physics Education," in 15th International Conference on Learning Analytics and Knowledge, LAK 2025, Association for Computing Machinery, Inc, Mar. 2025, pp. 620–630. doi: 10.1145/3706468.3706550.

[31] N. Dehbozorgi and M. T. Kunuku, "From Reflection to Insight: Using LLM to Improve Learning Analytics in Higher Education," in 2025 ASEE Annual Conference & Exposition, Montreal, Quebec, Canada, 2025. doi: 10.18260/1-2--56616.

[32] D.-K. & Mah and M. Egloffstein, "A Co-Creative Approach for AI-Enhanced Instructional Design Combining Generative Artificial Intelligence and Learning Analytics," Journal of Applied Instructional Design, vol. 14, no. 3, 2025, doi: 10.59668/2223.22719.

[33] J. Liu, S. Li, and Q. Dong, "Collaboration with Generative Artificial Intelligence: An Exploratory Study Based on Learning Analytics," Journal of Educational Computing Research, vol. 62, no. 5, pp. 1234–1266, Sep. 2024, doi: 10.1177/07356331241242441.

[34] M. Li, W. Qin, Z. Tang, X. Fang, T. He, and X. Cao, "Automating SSRL Detection in Asynchronous OCL via LLMs," in 2025 7th International Conference on Computer Science and Technologies in Education, CSTE

2025, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 548–551. doi: 10.1109/CSTE64638.2025.11092245.

[35] S. Ramanathan, L. A. Lim, N. R. Mottaghi, and S. Buckingham Shum, "When the Prompt becomes the Codebook: Grounded Prompt Engineering (GROPROE) and its application to Belonging Analytics," in 15th International Conference on Learning Analytics and Knowledge, LAK 2025, Association for Computing Machinery, Inc, Mar. 2025, pp. 713–725. doi: 10.1145/3706468.3706564.

[36] J. Lee, A. J. Gallard, G. Grigoryan, M. Yilmaz Soylu, and R. Desai, "Leveraging Large Language Models to Detect and Summarize Cognitive Presence in Online Discussion Forums," in 2025 IEEE Digital Education and MOOCS Conference (DEMOcon), Boca Raton, FL, USA, 2025, pp. 1–6. doi: 10.1109/DEMOcon65705.2025.11282579.

[37] A. Nguyen, B. Dang, F. Gul, and L. Huynh, "Mapping the Knowledge Construction Process Through Interactions with an Embodied GenAI Agent in Mixed Reality," in Proceedings - 25th IEEE International Conference on Advanced Learning Technologies, ICALT 2025, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 212–216. doi: 10.1109/ICALT64023.2025.00066.

[38] R. Whitehead, A. Nguyen, and S. Järvelä, "Utilizing Multimodal Large Language Models for Video Analysis of Posture in Studying Collaborative Learning: A Case Study," Journal of Learning Analytics, vol. 12, no. 1, pp. 186–200, Mar. 2025, doi: 10.18608/jla.2025.8595.

[39] Y. Jin et al., "Chatting with a Learning Analytics Dashboard: The Role of Generative AI Literacy on Learner Interaction with Conventional and Scaffolding Chatbots," in In Proceedings of the 15th International Learning Analytics and Knowledge Conference (LAK '25), New York, USA: Association for Computing Machinery, 2025, pp. 579–590. doi: https://doi.org/10.1145/3706468.3706545.

[40] X. Ochoa, X. Huang, and Y. Shao, "Exploring the Potential of Generative AI to Support Non-experts in Learning Analytics Practice," Journal of Learning Analytics, vol. 12, no. 1, pp. 65–90, Mar. 2025, doi: 10.18608/jla.2025.8615.

[41] L. Yan et al., "The effects of generative AI agents and scaffolding on enhancing students' comprehension of visual learning analytics," Comput. Educ., vol. 234, Sep. 2025, doi: 10.1016/j.compedu.2025.105322.

[42] G. Li, C. Tang, L. Chen, D. Deguchi, T. Yamashita, and A. Shimada, "LLM-Driven Ontology Learning to Augment Student Performance Analysis in Higher Education," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Science and Business Media Deutschland GmbH, 2024, pp. 57–68. doi: 10.1007/978-981-97-5498-4_5.

[43] U. Pujianto, A. K. Nanda, and D. Sutaji, "Predicting Student Academic Performance Based On A Neural Network Classification Model Of Discussion Activity With ChatGPT," Institute of Electrical and Electronics Engineers (IEEE), Nov. 2025, pp. 1–6. doi: 10.1109/iceeie66203.2025.11252479.

[44] A. Berr, S. Leelaluk, C. Tang, L. Chen, F. Okubo, and A. Shimada, "Educational Data Analysis using Generative AI," in Proceedings of LAK 2024 Workshops, Kyoto, Japan, Mar. 2024.

[45] H. Li and Z. Liu, "An Intelligent Educational System: Analyzing Student Behavior and Academic Performance Using Multi-Source Data," Electronics (Switzerland), vol. 14, no. 16, Aug. 2025, doi: 10.3390/electronics14163328.

[46] G. Kostopoulos, M. Tsiakmaki, and S. Kotsiantis, "Benchmarking Statistical and Deep Generative Models for Privacy-Preserving Synthetic Student Data in Educational Data Mining," Algorithms, vol. 19, no. 1, p. 39, Jan. 2026, doi: 10.3390/a19010039.

[47] N. T. K. Son, N. H. Quynh, and B. T. Minh, "Refining Graduation Classification Accuracy with Synergistic Deep Learning Models," Cybernetics and Information Technologies, vol. 25, no. 2, pp. 131–151, Jun. 2025, doi: 10.2478/cait-2025-0016.

[48] A. Böttcher, S. Dünnebeil, and V. Thurner, "Applying Generative AI to Analyze Statements from Student Surveys," in International Conference on Higher Education Advances, Universidad Politecnica de Valencia, 2025, pp. 1291–1298. doi: 10.4995/HEAd25.2025.20043.

[49] A. Slim et al., "AI-Driven Student Data Analytics: Leveraging Generative AI for Institutional Decision-Making," in IEEE International Conference on Electro Information Technology, IEEE Computer Society, 2025, pp. 322–329. doi: 10.1109/eIT64391.2025.11103654.

[50] A. Chen et al., "Unpacking help-seeking process through multimodal learning analytics: A comparative study of ChatGPT vs Human expert," Comput. Educ., vol. 226, Mar. 2025, doi: 10.1016/j.compedu.2024.105198.

Correlation-Aware Hybrid Feature Selection for Efficient Machine Learning-Based Intrusion Detection in Smart Home IoT Networks

Samar M. Alqhtani
Department of Information Systems, College of Computer Science and Information Systems,
Najran University, Najran, 61441, Saudi Arabia

***Abstract*—With the emergence of smart homes based on the Internet of Things, network-based attacks have become more prevalent and require effective and lightweight intrusion detection (ID). In this study, a Correlation-Aware Hybrid Feature Selection (CAHFS) framework for smart home IDS is proposed. The proposed approach fuses correlation analysis and Mutual Information (MI) and Random Forest (RF) feature importance to eliminate redundant and less important features and keep only those features that contribute most to the classification. To evaluate the different feature-selection strategies, four machine-learning models were tested: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF) and HistGradientBoosting. Among the classifiers, HistGradientBoosting provides the most stable results out of the assessed models. The proposed CAHFS feature set, using only 16 features, yielded 91% accuracy, 91% precision, 90% recall, 90% F1-score, and a ROC-AUC of 0.96. It was also able to attain an accuracy level of about 95% specificity with an overall error rate of 8.9%, which provides a good balance between detecting the attacks and avoiding false alarms. The comparison with other feature-selection methods demonstrated that CAHFS managed to decrease the number of features from 22 to 16, with the same or similar classification results. Overall, the findings indicate that CAHFS can offer a more compact and informative feature set to be used in machine-learning-based intrusion detection in smart-home networks, which may benefit networks with limited computational resources.**

***Keywords*—Smart home; intrusion detection; machine learning-feature representation; correlation-aware hybrid feature selection**

I. Introduction

The Internet of Things (IoT) has transformed the way people today live in their homes by bringing many different types of devices together, from cameras to sensors, smart devices to lighting systems, voice assistants to access-control systems. They often use communication technologies like Wi-Fi, Bluetooth, and Zigbee, and may even connect with cloud-based services and mobile apps [1][2]. This connectivity is very convenient, automated, and accessible, but also increases security threats with the number of connected points that can be attacked [3][4]. Smart home systems can be targeted by attacks like unauthorized access, information theft, device manipulation, and network reconnaissance, which compromise their confidentiality, integrity, and availability. Fig. 1 illustrates how connected IoT devices, communication technologies, automation, and remote services provide convenience and accessibility while creating a larger networked environment that requires effective security monitoring.

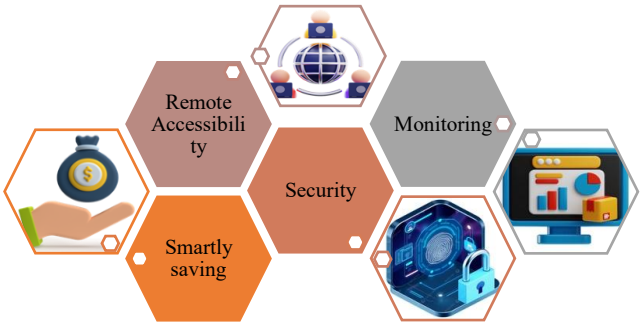


Fig. 1. Key benefits of smart home system.

A smart home can be viewed from a security perspective as a continual process that involves devices creating and sharing data, network traffic gathering and tracking, suspicious activity analysis, and necessary security measures. In the process, an Intrusion Detection System (IDS) is placed to monitor network traffic and detect whether it is normal traffic or if a possible attack is being made. In the context of smart homes, the IDS becomes even more crucial due to the wide range of capabilities and communication methods among IoT devices. A lot of these devices are also constrained in their computing and storage capacity and connected to the network for extended periods of time. Thus, an effective IDS should be able to detect the attacks efficiently without adding unnecessary complexity [5][6].



Fig. 2. Functional benefits of intrusion detection system in smart-home environment.

Fig. 2 summarizes the role of IDS in monitoring network traffic, identifying suspicious activities, distinguishing legitimate and malicious behavior, and supporting timely security response.

Traditional IDS methods rely on signature- and rule-based detection, which are able to detect attacks that have a known pattern. However, they may not be as effective when facing new or changed attack behaviors. Machine learning (ML) methods have therefore received a lot of attention in terms of IoT intrusion detection, as they can learn patterns from network traffic and then identify whether the traffic is legitimate or malicious, based on the patterns learned. In the past, various IDS configurations have been studied, such as centralized, distributed, edge, and fog systems, as well as various machine learning and deep learning algorithms [7][8].

Even with these advances, selecting an appropriate classifier is just part of the intrusion detection problem. The effectiveness and efficiency of the resulting detection model are also directly affected by the representation of the input data. Features in smart home traffic datasets may include many similar features, and there may be others that do not help to differentiate between legitimate traffic and attack. Redundant variables or variables that contribute only a marginal amount of information to the learning problem can make the problem unnecessarily complex and may lead to the creation of an unnecessarily high-dimensional model. Removing features due to correlation can also remove variables that carry useful information for attack classification, on the other hand. Another limitation may arise when evaluating feature relevance based on a single criterion, as different approaches may focus on different aspects of the usefulness of a feature. This shows that there is a need for a feature selection method that takes into account feature redundancy and usefulness to the intrusion detection problem [9].

To address the issue, this study Correlation-Aware Hybrid Feature Selection (CAHFS) model for smart home intrusion detection. The proposed method is intended to remove unnecessary and repetitive features from network traffic data while keeping the information needed to distinguish between normal and attack activity. To assess its effectiveness, the selected feature set is tested with four machine learning models, namely LogisticRegression, DecisionTree, RandomForest, and HistGradientBoosting. The comparison is based on standard classification metrics and error detection analysis using an independent test set.

The study does not propose a new classification algorithm; rather, it focuses on improving the input representation through a correlation-aware hybrid feature selection strategy. The proposed CAHFS model jointly considers feature redundancy and predictive relevance by combining correlation analysis, mutual information, and random forest importance. This provides a broader assessment of feature usefulness than relying on a single selection criterion. The resulting compact feature representation is subsequently evaluated using ML classifiers to determine whether dimensionality reduction can be achieved without substantially compromising the intrusion detection system.

To systematically evaluate the effectiveness of the proposed model, this study addresses the following Research Questions (RQ): -

RQ-1: How does the proposed CAHFS feature representation affect the performance of various ML classifiers for smart-home intrusion detection compared with alternative feature selection strategies?

RQ-2: Which ML classifier provides the most effective balance between detection performance and classification errors when using the proposed CAHFS feature representation?

The following are the main contributions of this study:

- A Correlation-Aware Hybrid Feature Selection (CAHFS) framework is proposed for a smart home intrusion detection system by combining Correlation-based redundancy analysis with complementary mutual information and random forest importance.
- The proposed approach reduces the original 22 features to 16 selected features, providing more compact input space while retaining discriminative information for intrusion detection.
- Four ML classifiers, namely (Logistic Regression, Decision Tree, Random Forest, and HistGradientBoosting), are systematically evaluated using multiple feature selection strategies under separate validation and independent test protocols.
- The classifiers are further assessed using classification metrics, confusion matrix, ROC AUC, and precision-recall analysis to examine both overall predictive performance and class-wise detection behavior.

The remainder of the study is divided into four sections. In Section II, the key research gap was identified. In Section III, the overall methodology adopted was discussed. In Section IV, the interpretation of the experimental analysis was defined and discussed. Finally, in Section V, necessary future directions and research conclusions were discussed.

II. Literature Review

Previous studies have shown that machine learning algorithms can enhance intrusion detection in IoT contexts by detecting network traffic patterns when they are normal or abnormal. Random Forest, XGBoost, and LightGBM prove to be good detection performance for ensemble methods. This study provides a comparison of these methods and presents improved efficiency, specifically 99.92% accuracy overall for the proposed LGB-IDS method [10]. Sophisticated attacks on IoT smart homes are not easily detected by traditional firewalls and intrusion detection systems. Machine learning, deep learning, and ensemble methods have recently been developed to boost detection accuracy and decrease the number of false positives, respectively. Combining these techniques in this study, CNN outperforms them on several datasets, obtaining an accuracy of around 99%, and the hybrid ensemble 99.8% [11]. Existing studies on intrusion detection for smart home systems with IoT typically use simulated data sets or small data sets, and do not consider the IoT devices themselves as an attack vector. This study proposes harmony search optimization of an Extra

Trees classifier for the detection of real-time traffic from 105 devices.

The proposed IDS obtained an accuracy of 99.87% in binary classification [12]. The emerging trend of smart home security systems based on the Internet of Things (IoT) demands efficient facial recognition models for resource-constrained devices. In the era of Internet of Things (IoT) enabled smart home security, efficient facial recognition models are required for resource-constrained devices. This study uses a network architecture named EfficientNet-B4 along with transfer learning and fine-tuning to enhance recognition with a small dataset. The model successfully reached an accuracy of 97% on collected images, whereas the EfficientNet-B4 model had a top-1 and top-5 accuracy of 82.9% and 96.4% [13]. Smart home devices of the IoT generation are becoming more susceptible to DDoS attacks, and it is essential to detect and mitigate such attacks in real time. This study suggests an SDN-based architecture that integrates the machine learning scheme for traffic classification on the IoT devices with the SNORT traffic analysis scheme for controller protection. SVM, Logistic Regression, Decision Tree and KNN were tested and among them Decision Tree has the maximum accuracy of 99% [14]. With IoT and machine learning, home security becomes intelligent, with the ability to detect threats in real time based on the information from cameras and movement sensors. Three models, namely Decision Tree, transfer learning, Logistic Regression, and SVM are evaluated in this study. The system proposed in this work was successful in reaching an accuracy of 99.2%, while Decision Tree and transfer learning performed better than SVM and Logistic Regression [15]. Anomaly detection is crucial in human activity for smart homes, especially when residents are elderly, and they may not be monitored constantly. The environment of multi-resident facilities adds to the complexities, and comprehensive anomaly datasets are limited. In this study, eight machine learning and two deep learning classifiers are compared, and it is concluded that GRU is the best when it comes to distinguishing abnormal and normal activities [16]. Smart home technologies have improved the convenience of the home, but opened up a great deal of security concerns, which are to be tackled by a good intruder detection system. The signature-based, anomaly-based, and hybrid IDS methods are the existing ones. Their interoperability with machine learning, IoT, and edge and cloud computing will be explored with a focus on dynamic, privacy-sensitive security solutions [17].

Research Gap: An overview of these studies reveals that most of the current studies are dedicated to the selection of an effective classifier and to the assessment of its detection performance, whereas relatively little work has been done on reducing the number of features without loss of useful information. The studies also process the information of features in various ways. In [10, 14, and 15] the different classification algorithms are explored mainly, while, in [11] the deep learning and ensemble methods are combined, and in [12] an optimization technique is applied in conjunction with an Extra Trees classifier. All of these studies look at the potential of various learning methods, but none of them explore feature redundancy and feature relevance in combination in a single feature-selection method. If removal of highly correlated features is undertaken alone, then it does not indicate which features, if any, from the remaining features have the most influence on target class prediction. This gap inspires the proposed CAHFS approach, which combines these two aspects: correlation analysis is applied to solve the redundancy problem, and Mutual Information and Random Forest feature importance are used to find informative features.

To overcome these problems, this study proposes a hybrid feature-selection method entitled CAHFS, which is based on the combination of correlation analysis, Mutual Information, and the feature importance of Random Forest. The goal is to eliminate features that are not needed while preserving features of the smart-home network traffic that are useful for intrusion detection. The resulting feature set is then evaluated on several machine-learning classifiers to assess the overall predictive accuracy of the classifiers and the error patterns.

III. METHODOLOGY

In this section, the methodology that has been adopted towards the development of the proposed smart home intrusion detection framework is presented. Data collection and data preprocessing are the first steps in the process, followed by exploratory data analysis to gain insights into the nature of the network traffic. The main methodological part is then a correlation-aware hybrid feature selection (CAHFS) that selects the most useful and least redundant features. Selected features are then applied to build and test various machine learning models under the same experimental conditions to compare them. As a general principle, the proposed workflow is designed to minimize the original feature space whilst preserving the information required for reliable intrusion detection and ensuring a fair comparison between the models evaluated. Fig. 3 illustrates the steps adopted.

A. Data Collection

The dataset used in this study was retrieved from a publicly available source [18] The dataset was captured under normal operation as well as from 7 different multi-stage attack scenarios that can arise in a smart home environment that hosts end devices of heterogeneous nature. The dataset was published in 14 January 2026 by Amrita Vishwa Vidyapeetham. The dataset comprised (178831 x 22).

The distribution of training and validation data for the two classes in the dataset is shown in Fig. 4. The test set contains 29,872 samples and was kept separate from the model development t process for final evaluation. Approximate Class ratio: 57348 normal samples, 91611 attack samples (1.591. The data set is moderately imbalanced as the attack samples outnumber the normal samples. To minimize the impact of this imbalance during the training of these models, SMOTE (Synthetic Minority Over-sampling Technique) was used to augment the training data. This would help to ensure that the minority class was represented and the likelihood of the models becoming biased towards the majority class was minimized.

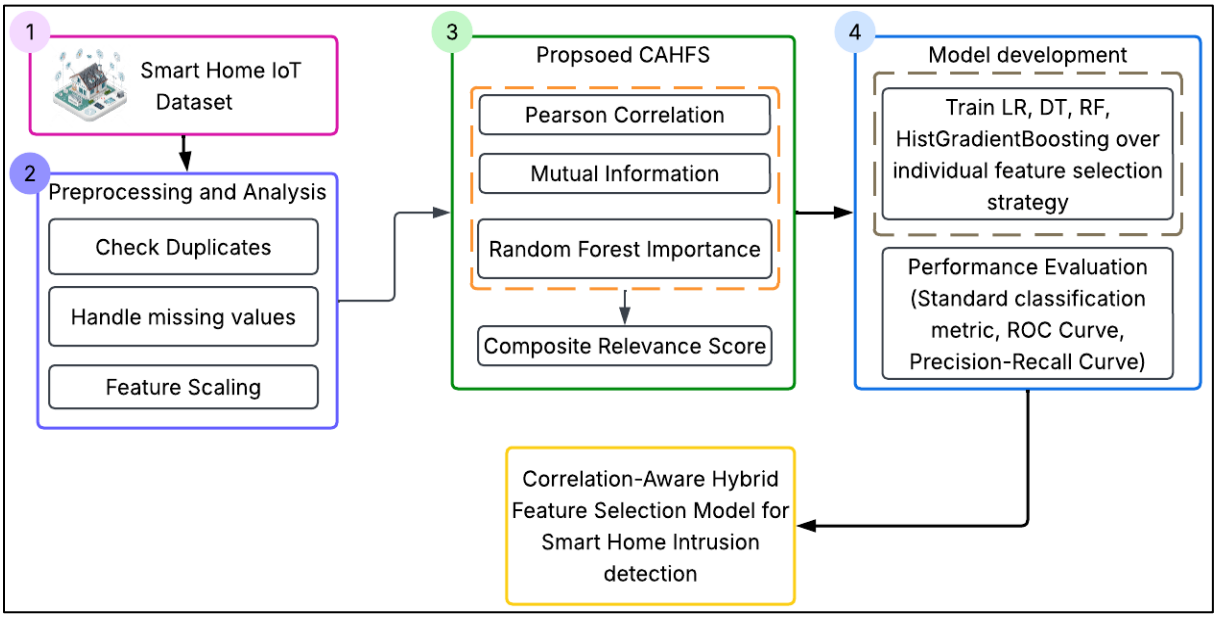


Fig. 3. Overall research pipeline.

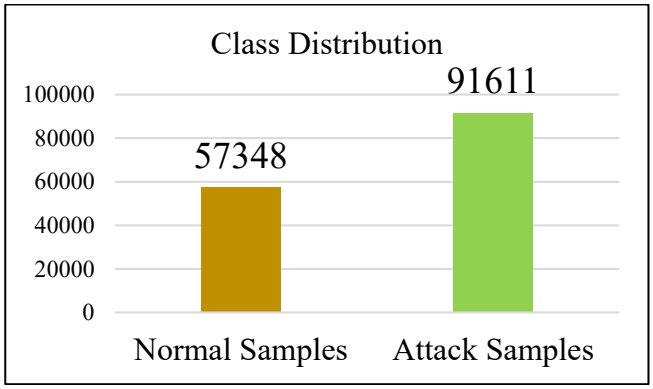


Fig. 4. Class distribution of training and validation data.

B. Preprocessing and Analysis

The raw network traffic is first cleaned and formatted to be consistent before creating the classification models. Records are duplicated and deleted as needed. All infinite values are replaced with missing values, and all values missing after that are processed in a preprocessing procedure that is tailored only to the training data. The training portion is then split into separate training and validation data sets with stratified sampling. This means that the number of normal instances is the same in both sets as the number of attack instances [19][20].

C. Proposed Correlation-Aware Hybrid Feature Selection (CAHFS)

In the context of smart home intrusion detection, feature selection plays a crucial role as network traffic datasets can frequently be large and include numerous flow and statistical features, some of which may lack discriminatory power or be redundant. This work presents a novel approach named Correlation-Aware Hybrid Feature Selection (CAHFS) to minimize the above redundancy and enhance the learning efficiency. CAHFS rates features based on three criteria: relevance to the intrusion class, importance in predicting the class, and redundancy with the other features selected. Mutual Information (MI) is a measure of target relevance, and Random Forest (RF) importance is a measure of predictive contribution. Pearson correlation is then applied to determine whether there is information that is shared between features. These criteria are aggregated into a score, with informative features getting a higher score and highly redundant features getting a lower score, with the criteria being correlation-aware. The procedure consists of four steps: 1) correlation analysis, 2) MI-based relevance estimation, 3) RF-based importance estimation, and 4) correlation-aware score fusion. This ranking is then applied to determine a small number of useful features to be used for the intrusion detection models.

Let the traffic in the dataset $\mathcal{D}$ be denoted by X and the available features be represented as d the corresponding intrusion classes be shown as Y_i Hence, the original feature space and training dataset is given by Eq. (1) and (2), respectively:

$$\mathcal{F} = \{X_1, X_2, \dots \dots X_d\} \tag{1}$$

$$\mathcal{D}_{t_r} = \{(X_i, Y_i)\}_{i=1}{}^{N} \tag{2}$$

1) Pearson correlation analysis: In this stage, the proposed CAHFS method examines the relationship between every pair of traffic features using Pearson correlation [21]. The correlation between features X_i and X_k is calculated by:

$$r_{jk} = \frac{\sum_{i=1}^{N}(X_{ij}-\overline{X_i})(X_{ik}-\overline{X_k})}{\sqrt{\sum_{i=1}^{N}(X_{ij}-\overline{X_i})^2}\sum_{i=1}^{N}(X_{ik}-\overline{X_k})^2} \tag{3}$$

A feature pair is highly correlated when $|r_{jk}| \geq 0.90$, thus the highly correlated feature set is defined as:

$$\mathcal{C} = \{(i,k): |r_{jk}| \geq 0.90\} \tag{4}$$

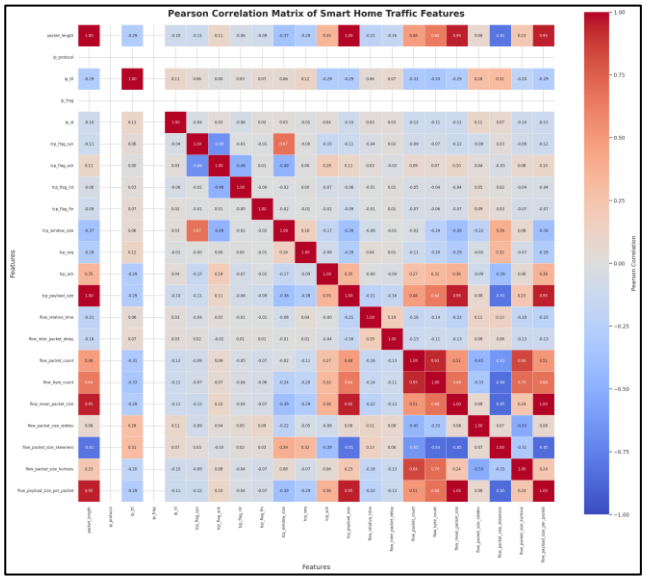


Fig. 5. Selected features using Pearson correlation method.

The training data was used to calculate the Pearson correlation matrix as illustrated in Fig. 5. It gives an idea of the linear relationships between the traffic features themselves. Correlation values near +1 or −1 signify a strong correlation, and values near 0 signify that there is little or no linear correlation. For this analysis, feature pairs having an absolute correlation coefficient value higher than 0.90 were considered highly correlated and redundancy was noted during feature selection.

This step identifies the potential redundancy among smart home traffic features.

2) Mutual Information (MI): After identifying the correlated features, mutual information is calculated to determine how strongly each feature is associated with intrusion class [22]. For feature X_j and Y represents the intrusion class, the MI score is calculated as:

$$\mathrm{I}_j = \mathrm{I}(X_j; Y) \tag{5}$$

A larger I_j indicates that the features provides more information about the intrusion class. The normalization of the MI score is defined as:

$$\overline{\mathrm{I}_j} = \frac{\mathrm{I}_j - \mathrm{I}_{min}}{\mathrm{I}_{max} - \mathrm{I}_{min}} \tag{6}$$

The scores of the input features are shown in terms of the Mutual Information with the intrusion class in Fig. 6. The higher the MI, the more information a feature contains about the class it represents and therefore the more value it is for discriminating between normal and attack traffic. Unlike Pearson correlation, which is primarily used in detecting linear relationships, Mutual Information can also detect nonlinear dependencies. This can be helpful in recognizing informative features that might not have been captured by the correlation analysis.

This step is important because MI can capture nonlinear relationships that may not be captured through the Pearson correlation method.

3) Random forest feature importance: A balanced Random Forest classifier is trained using the training dataset. The predictive contribution of each feature is obtained from the trained ensemble [22][23]. For feature X_j, its random forest importance is defined by:

$$\mathcal{F}_j = RFimportance(X_j) \tag{7}$$

The normalization score of $\mathcal{F}_j$ is expressed by:

$$\overline{\mathcal{F}_j} = \frac{\mathcal{F}_j - \mathcal{F}_{min}}{\mathcal{F}_{max} - \mathcal{F}_{min}} \tag{8}$$

A larger $\mathcal{F}_j$ indicates that feature contributes more strongly to the predictive decisions for Random Forest.

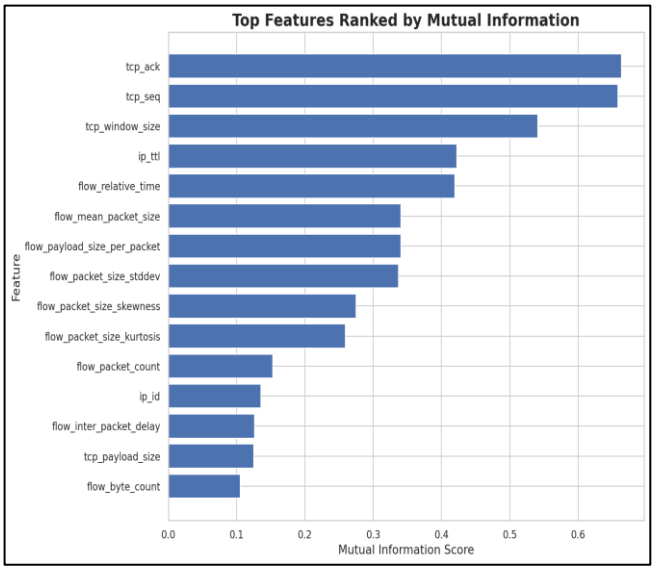


Fig. 6. Top features ranked by MI method.

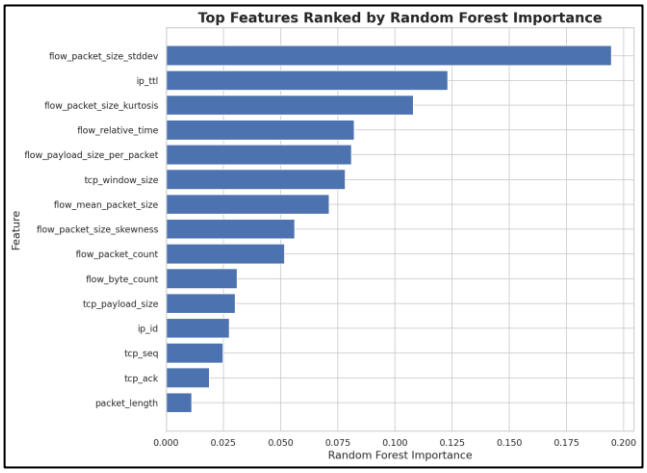


Fig. 7. Random forest importance features.

The importance scores obtained by the Random Forest model using the training data is shown in Fig. 7. The larger the importance score, the more the feature influences the prediction of the model; the smaller the score, the less the feature influences the model. This ranking is another perspective on feature usefulness in addition to Mutual Information and correlation

analysis, since Random Forest can discover nonlinear patterns in the data

4) Proposed CAHFS-based composite score: For feature $\mathcal{S}$ represent the set of features that have already been selected. For a candidate feature X_j its correlation redundancy with the selected features is defined as:

$$\mathcal{R}_j = \max_k \in \mathcal{S}[r_{jk}] \quad (9)$$

Finally, the normalized MI and RF scores are then combined using equal weights to obtain the final composite score for each feature:

$$\mathcal{S}_j^{CAHFS} = \frac{1}{2}\bar{\mathrm{I}}_j + \frac{1}{2}\bar{\mathcal{F}}_j \quad (10)$$

where, $\bar{\mathrm{I}}_j$ represents the normalized MI score, while $\bar{\mathcal{F}}_j$ denotes the normalized Random Forest Importance score. $\mathcal{S}_j^{CAHFS}$ shows the final CAHFS composite score of features X_j.

This means that a feature is preferred when it is informative about the intrusion class, useful for prediction, and not excessively redundant with features that have already been selected.

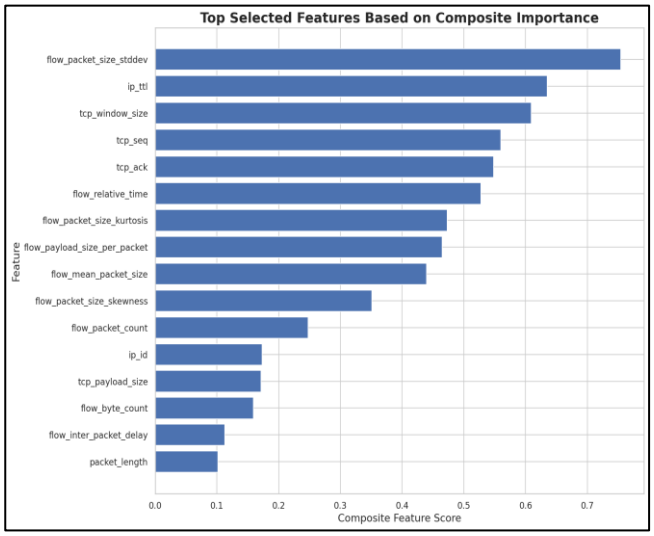


Fig. 8. Composite importance of selected features.

The final feature ranking derived by CAHFS (after merging the relevance scores obtained from Mutual Information with feature importance scores from Random Forest) is presented in Fig. 8. The higher the combined scores for each feature, the closer it will be at the top of the ranking, and the closer it is to the top, the more likely it is to be selected according to the predetermined retention criteria. The reduced feature set is then fed into all the classification models, and the experimental conditions are kept the same during the evaluation.

The final number of features was rounded to the nearest whole number and 16 features remained out of an initial set of 22 features. For consistency, the same retention threshold was used in all the feature-selection experiments. In this study, there is no systematic testing of different retention levels, and this is identified as a limitation and is suggested for further work to test the effects that various sizes of feature sets have on the classification performance-dimensionality tradeoff.

Algorithm-1: Proposed CAHFS Method

Input: training feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, training labels $\boldsymbol{Y} \in \{0,1\}^N$, correlation threshold $\tau = 0.90$, Feature retention ratio $r = 0.70$

Output: selected Feature Subset $\mathcal{S}$

Start:

1. Load $X \in \mathbb{R}^{N \times d}$ and $\in \{0,1\}^N$
2. Compute Pearson correlation Matrix $\mathcal{R}_j$ using Eq. (3)
3. Identify highly correlated feature pairs using Eq. (4)
4. Compute Mutual Information between each feature and target labels using Eq. (5)
5. Train a Random Forest classifier using X and Y
6. Extract Feature Importance using Eq. (7)
7. Normalize the MI and RF scores using Min-Max normalization
8. Calculate the final composite CAHFS score using equal weights 0.50 for MI and RF, as defined in Eq. (10)
9. Rank all features in descending order according to their CAHFS score.
10. Determine the number of retained features using the predefined retention ratio:
11. $N_{SELECTED} = \lceil rN_{features} \rceil$
12. Select Top-ranked features
13. Return $\mathcal{S}$

End

The proposed Correlation-Aware Hybrid Feature Selection (CAHFS) procedure is described in Algorithm 1. It first calculates the Pearson correlation matrix for the training data and finds the pairs of features that are highly linearly correlated. The relevance and importance of each feature with respect to intrusion detection is then measured separately using Mutual Information and Random Forest. The scores from both measures have been normalized; the scores of the two measures are then pooled together with equal weights to get the overall relevance score. This combined score is used to rank the features and the top 70% are selected as the final set of features. These selected features are then given to the machine-learning models, which would then be trained and evaluated in turn.

D. Model Development

Four machine-learning models are used to evaluate the selected 16 features. The models are linear classification, single tree learning, bagging, and gradient boosting.

1) Logistic regression: As a linear baseline to investigate whether the selected features can distinguish between normal and attack traffic with a simple decision boundary [24].

2) Decision tree: Recursively partitions the feature space based on decision rules, thus enabling the model to learn nonlinear attack patterns [25].

3) Random forest: Aggregates several decision trees to achieve greater stability and to pick up more interactions among the features [26].

4) HistGradientBoosting: An efficient model, based on the gradient boosting framework, to learn non-linear relationships

using the histogram technique and to yield an excellent model with the selected feature set [27].

TABLE I. HYPERPARAMETERSARAMETERS

Models	Parameters	Description
LR	C	0.1
	Penalty	L2
RF	n_estimators	100
	Max_depth	None
	Min_samples_split	2
	Min_samples_leaf	1
DT	Max_depth	10
	Min_samples_split	2
	Min_samples_leaf	1
	Entropy	
HistGradientBoosting	Learning Rate	0.1
	Max_iteration	150
	Max_leaf_nodes	31
	12_regularization	0.1

The parameter values used for the four machine-learning classifiers in the experiments are shown in Table I. These settings were kept constant for all feature-selection tests, enabling the comparison of the different models under the same conditions. The validation set was used during the development of the model for evaluation and tuning, and the test set was not used until the final evaluation. These details have been added to the revised manuscript to adequately describe the experimental setup and to aid in the reproduction of the study. Logistic Regression: L2 regularization (C=0.1). Random Forest employed 100 trees and did not specify the maximum depth of the trees. The maximum depth was 10 for the Decision Tree, and entropy was chosen as the splitting criterion. The model, named HistGradientBoosting, was trained with a learning rate of 0.1, 150 boosting iterations, 31 maximum leaf nodes, and L2 regularization of 0.1. The same settings were used in all experiments to ensure as far as possible, a consistent comparison among the various feature subsets.

IV. RESULTS AND DISCUSSION

Experiments were all performed in Python, and all the usual machine-learning libraries were used for data pre-processing, feature selection, model development, and evaluation of results [28]. The data set was split into training, validation, and test sets to maintain consistency and an unbiased experimental procedure. All the parameters of the preprocessing were set by the training data alone, and SMOTE was used to solve the class imbalance problem in the training set. Feature selection was also performed based on the training data only. Models were developed using the training set, and the validation set was used to select and tune the model parameters. The test set was held entirely apart from the entire development process and used only for the final evaluation. The data used was made up of 119,167 training samples, 29,792 validation samples, and 29,872 test samples as seen in Table II. By not changing the test set during model development, the final results were able to give a more accurate indication of the degree of generalization of the models evaluated.

TABLE II. DISTRIBUTION OF SAMPLES

Dataset Partition	Samples	Percentage
Training	(119167, 16)	66.64%
Validation	(29792, 16)	16.70%
Testing	(29872,16)	16.66%

TABLE III. PERFORMANCE OF ML CLASSIFIERS USING PROPOSED CAHFS

ML models	Accuracy	Precision	Recall	F1 score
LR	62.71%	56.69%	51.80%	54.14%
RF	80.04%	82.29%	66.76%	74.97%
DT	85.61%	84.70%	80.69%	82.65%
HistGradientBoosting	91.03%	92.78%	85.55%	89.02%

Table III represents the performance of ML classifiers using the proposed CAHFS. Amomng classifiers HistGradientBoosting classifer outperforms acorss all metric as compared to other models by achiving an of accuracy (91.03%), precision (92.78%), recall (85.55%), and f1 score (89.02) its superiro performance can be attributed to its nonline fetaure inetractions and complex decision boundries, which are improtant for distinuguishing normal and attack traffic. Followed by DT, which exhibits accuracy (85.61%), precision (84.70%), recall (80.69%), and F1 score (82.65%). The RF classifier performs average, obtaining an accuracy (80.04%), precision (82.29%), recall (66.76%), and F1 score (74.97%). The LR classifer perfomed poor by achieving the lowest accuracy (62.71%), precision (56.69%), recall (51.80%), and F1 score (54.14%), likely because its linear decision boundary is less effective for the complex relationship present in the selected features.

TABLE IV. CLASS-WISE ERROR CHARACTERISTICS AND SPECIFICITY OF THE EVALUATED ML CLASSIFIER ON UNSEEN TEST SET

Model	True Negative	False Negative	True Positive	False Positive	Specificity	Overall error rate
LR	12160	6116	6574	5022	0.707	0.372
DT	15333	2451	10239	1849	0.892	0.143
RF	15437	4218	8472	1745	0.898	0.199
HistGradientBoosting	16337	1834	10856	845	0.950	0.089

Table IV summarizes the errors of the evaluated classifiers on the test set that had not been included in the development of the classifiers. HistGradientBoosting had the best performance of all four models. It had the lowest false-positive rate (4.92%)

and the lowest false-negative rate (14.45%), the highest specificity (95.08%), and the lowest overall error rate (8.97%). This in turn allowed the model to successfully prevent false alarms and missed attacks on normal traffic**.** DT gave the next best results of a 19.31% false-negative rate, 10.76% false-positive rate, and 14.39% error rate. The error rate for Random Forest was higher (19.96%) due to a higher false-negative rate of 33.24%, even though it had a slightly lower false-positive rate (10.16%). Logistic Regression did the worst with false-positive and false-negative rates of 29.23% and 48.20%, respectively. In general, the results obtained indicate that HistGradientBoosting gave the best error profile, especially with the number of false alarms and the number of missed intrusion cases.

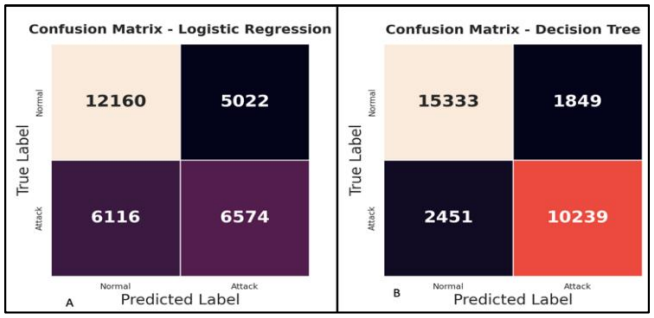


Fig. 9. Confusion matrix for LR and DT.

Fig. 9 illustrates the performance of LR and DT models over unseen test samples. In Fig. 9 (A), the confusion matrix for LR is shown, where the model exhibits lower performance, correctly classifying (18734) samples out of (29872), while (11138) samples were misclassified. Conversely, in Fig. 9(B), the performance of the DT classifier is shown. From the total samples (29872), 25572 samples were correctly predicted; only 4300 samples were misclassified.

Fig. 10 represents the performance of RF and HistGradientBoosting classifiers over test samples. In Fig. 10(A) the performance of RF model is depicted where the model correctly classifed (23909) samples from total (29872) while 5963 samples were wrongly classified. On the other hand, Fig. 10(B) shows the confusion matrix of the HistGradientBoosting classifier, where the classifier remarkably classified (27193) samples with only 2679 misclassifications.

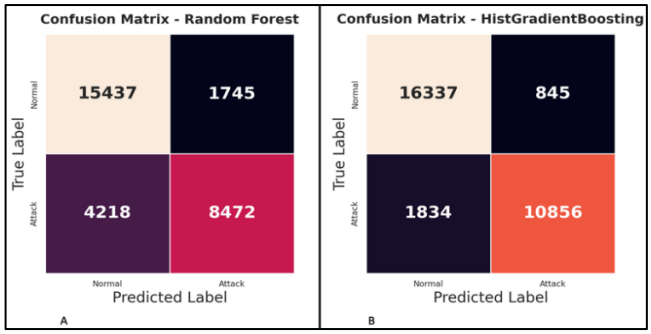


Fig. 10. Confusion matrix for RF and HistGradientBoosting.

The ROC curves for the four ML classifiers using the sets of features selected by the proposed CAHFS method are presented in Fig. 11. HistGradientBoosting (0.964) performed best with a ROC-AUC value, followed by RF (0.910) and DT (0.848). LR had the lowest ROC-AUC of 0.669, which clearly displayed its separation capability between two classes. The results show that HistGradientBoosting performed better than the other models for distinguishing between the desired classes for a variety of classification thresholds. It may be because its higher ROC-AUC allows it to learn the nonlinear relationship and interaction among the selected features. The trees are constructed on a sequential basis, with each new tree correcting the errors made by previous trees, which can enhance the overall classification performance. The opposite of this is true for the lower ROC-AUC value achieved by LR, which indicates that there is not a good linear association between the features used and the class labels.

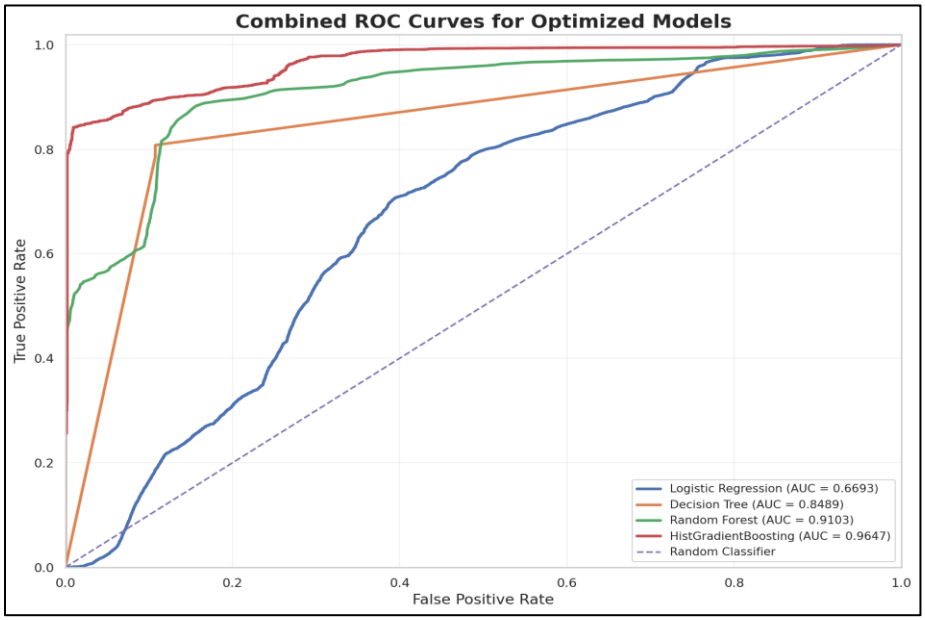


Fig. 11. ROC Curve of ML classifier using proposed CAHFS.

TABLE V. COMPARATIVE EVALUATION OF ML CLASSIFIERS USING ALTERNATIVE AND PROPOSED FEATURE-SELECTED STRATEGIES

Feature Methods	No Features	Models	Accuracy	Precision	Recall	F1 score	ROC-AUC
All	22	LR	63%	60%	61%	60%	0.673
		DT	86%	84%	83%	83%	0.855
		RF	80%	82%	67%	74%	0.916
		HistGrdaientBoosting	89%	90%	85%	87%	0.955
Pearson	18	LR	63%	59%	49%	53%	0.647
		DT	86%	86%	82%	84%	0.863
		RF	82%	87%	68%	76%	0.933
		HistGrdaientBoosting	92%	95%	85%	90%	0.9605
MI Only	16	LR	62%	56%	51%	54%	0.669
		DT	86%	85%	81%	83%	0.855
		RF	80%	83%	66%	74%	0.914
		HistGrdaientBoosting	90%	92%	85%	88%	0.964
RF Only	16	LR	62%	56%	51%	54%	0.669
		DT	86%	85%	81%	83%	0.855
		RF	80%	83%	66%	74%	0.914
		HistGrdaientBoosting	89%	89%	85%	87%	0.963
Proposed CAHFS	16	LR	62%	60%	59%	59%	0.66
		DT	80%	82%	79%	80%	0.85
		RF	85%	84%	80%	81%	0.91
		HistGrdaientBoosting	91%	91%	90%	90%	0.96

The performance of the four machine-learning classifiers in the three feature selection approaches (conventional feature selection, the proposed CAHFS approach, and the full set of features) is compared in Table V. Overall, the best results are obtained with HistGradientBoosting for varying feature configurations. It was able to reach an accuracy of 89%, an F1-score of 87%, and a ROC-AUC of 0.955 when using all 22 features. After feature selection using Pearson, accuracy and F1 score were 92% and 90%, respectively, while the number of selected variables decreased to 18. Further, the number of features was reduced to 16 variables by Mutual Information or RF selection, which yielded similar high ROC-AUC values of 0.964 and 0.963, respectively, in HistGradientBoosting. The proposed CAHFS method also generated competitive outcomes using just 16 features. With the HistGradientBoosting model, it achieved 91% accuracy, 90% recall, 90% F1 score, and a ROC-AUC of 0.960. The results indicate that the CAHFS can eliminate less useful variables without compromising the important information for classification. That is, the feature set is reduced, but information is still retained and input representation is compact. The superior results of HistGradientBoosting over LR, DT, and RF suggest that there are possible nonlinear relationships or interactions between the features in the data. A gradient-boosting approach is better suited than a linear classifier or traditional tree-based models to detect such patterns.

CAHFS provides slightly less accurate and precise classification, but with two fewer features. This indicates that the main advantage of CAHFS lies in its ability to provide a more compact and efficient feature representation rather than in achieving the highest accuracy. The overall results indicate that the reduction of input features while preserving the information required for reliable classification is possible with CAHFS.

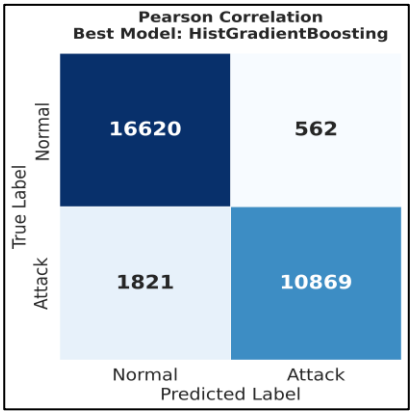


Fig. 12. Confusion matrix for the best-performing model.

The confusion matrix for the best-performing experimental setup (Pearson Selected-HistGradientBoosting) is presented in Fig. 12. The matrix shows that the model was able to correctly predict the majority of the normal and attack samples, with a relatively small number of false predictions. This is in line with the very good accuracy and F1-score of the model. The results also indicate that some redundant information was removed by using the features selected by the Pearson method without the loss of the features important for distinguishing between the two classes. This led to HistGradientBoosting having a good classification result on fewer input variables.

The ROC curve of the Pearson-selected HistGradientBoosting configuration is presented in Fig. 13. The curve remains in the upper left part of the plot, suggesting that the model has the ability to distinguish between normal and attack samples for various decision thresholds. The corresponding ROC-AUC value is also a sign of its overall discriminatory power. The ROC-AUC scores are higher for classifiers that can achieve a good balance between true positives and false positives for a range of thresholds. This indicates that the feature set chosen has the information necessary for the effective detection of intrusions.

The precision-recall curve of the Pearson Selected-HistGradientBoosting is shown in Fig. 14. The curve depicts the changes in precision and recall with varying classification thresholds. The analysis is particularly relevant in this context to investigate the ability of the model to find attack samples and keep the rate of incorrect positive predictions under control. The accuracy of the classifier's performance in the precision-recall graph shows that it can identify a significant percentage of attack samples with only a modest number of false alarms. In addition to the ROC analysis, this gives a better sense of the model's positive class identification ability.

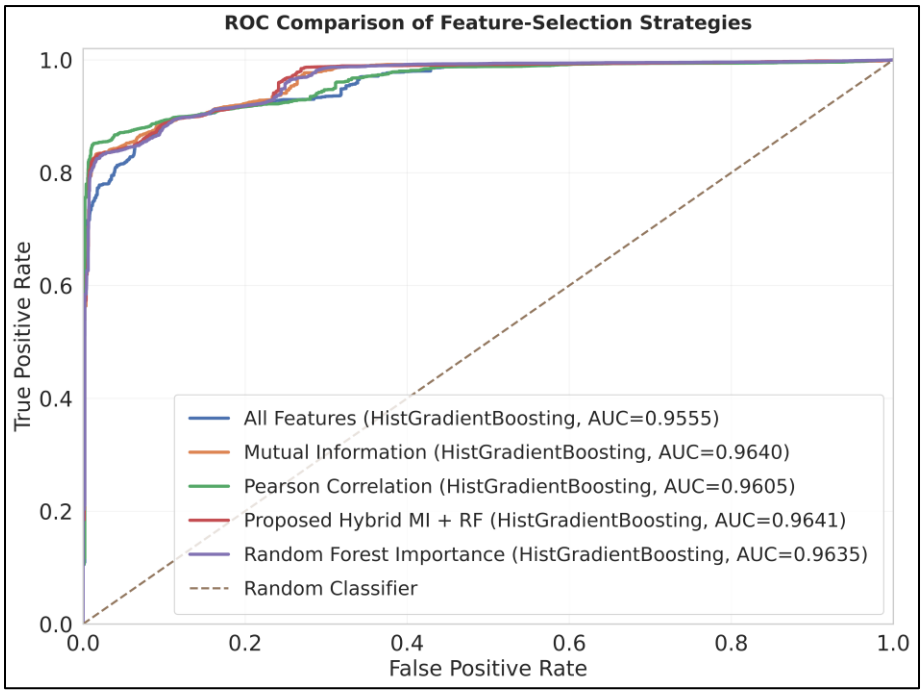


Fig. 13. ROC curve analysis of the best-performing model.

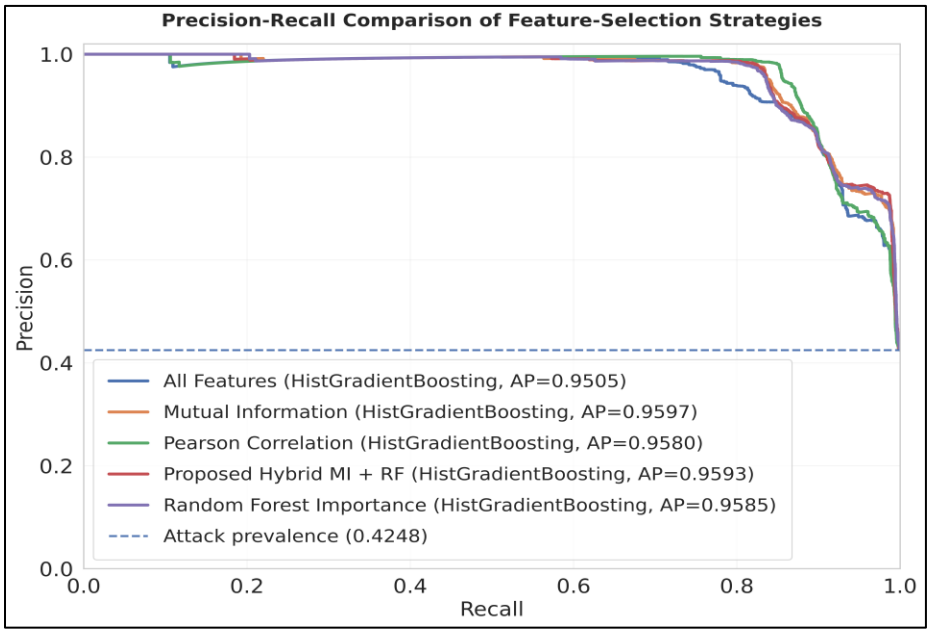


Fig. 14. Precision-recall curve of the best-performing model.

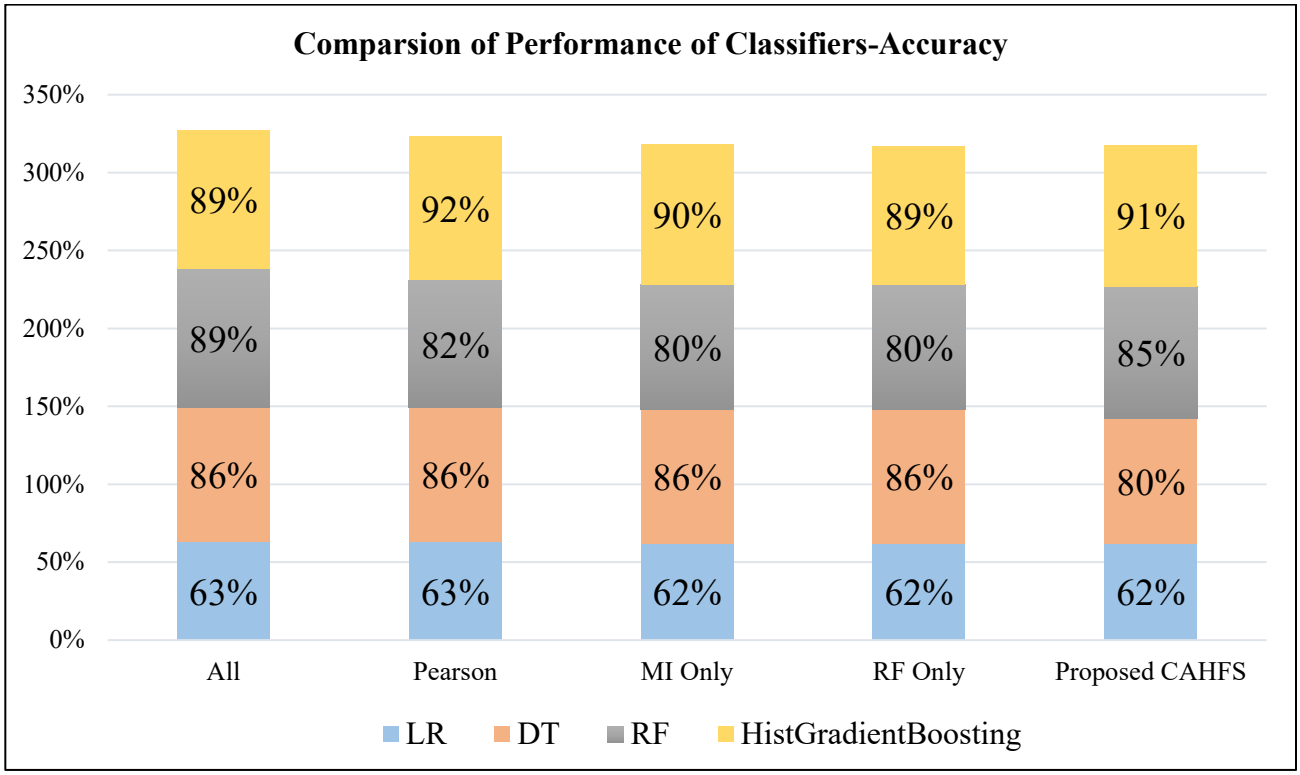


Fig. 15. Comparison of model performance across accuracy.

The results of the machine-learning models evaluated are compared in Fig. 15 under the various feature-selection strategies. Results indicate that the effect of feature selection was different for the classifiers. The Logistic Regression model performed relatively poorly over the tested feature sets while HistGradientBoosting performed the best. It is also observed from the comparison that a decrease in the number of features did not necessarily mean that the classification results are worse. In a few instances, the elimination of less informative or redundant variables resulted in better model performance. The proposed CAHFS method also yielded competitive results, with only 16 features, demonstrating that the selection of a set of features can provide a compact representation without loss of information relevant to intrusion detection.

V. Discussion

The experimental results demonstrate that the proposed correlation-aware hybrid feature selection (CAHFS) framework is able to reduce the input feature number without significant degradation in ID performance. Overall, HistGradientBoosting performed best when compared to the other classifiers, with an accuracy, precision, recall and F1-score of 91%, 91%, 90%, and 90%, respectively, as reported in Table III. The values indicate that the model could successfully detect the attacks and had a low rate of false alarms. Logistic Regression, however, did not do as well – indicating that the patterns in the logistic regression model between the features of the network traffic and the attack classes are not well described using only a linear decision boundary.

The difference is also evident in the results of the tests reported in Table IV. HistGradientBoosting had the best overall error rate of 8.9% and the highest specificity of 95.0%. It also made fewer misclassifications than the other classifiers. This is crucial for a smart home intrusion detection system, as false alarms can make a security system hard to manage, and missed attacks can let harmful activity slip through the cracks. These results are confirmed by the confusion matrices displayed in Fig. 9 and 10, which illustrate the accuracy of the models at predicting the test samples for the two types of classification outlined above.

Regarding RQ1, the results indicate that CAHFS was able to classify the same number of features as the original with a modest drop in classification accuracy. The accuracy was slightly higher with Pearson-based selection, but CAHFS still obtained a ROC-AUC of 0.96 using fewer features. This indicates that the proposed hybrid technique was able to remove less useful information and retain the features that are important for distinguishing between normal and attack traffic.

This can be observed in the ROC analysis shown in Fig. 11. For the ROC-AUC scores, HistGradientBoosting was able to distinguish normal from attack traffic for most classification thresholds, with an AUC score of approximately 0.96. A possible explanation for this better performance is that it can derive non-linear relationships between features. The interactions among multiple variables may be present in the network traffic produced by a smart-home environment, and may not be well captured by a linear model.

From results presented in Table V, the selection of the feature-selection method also had a significant impact on the results of the classification. The best accuracy, 92%, was obtained by the Pearson-based selection. Despite this, the

original 22-feature representation was reduced to just 16 features, yielding competitive results of 91% accuracy, 90% recall, 90% F1 score, and a ROC-AUC of 0.96. Fig. 12- 15 show the individual results of the model performance, including, among others, the confusion matrix, the ROC behavior, the precision-recall relationship, and a comparison of the various feature-selection strategies.

For RQ2, the overall best performance was achieved by the HistGradientBoosting classifier. This ROC-AUC, specificity (around 95%), and overall error rate (around 8.9%) suggest that it performed better at separating the attack traffic from the normal traffic and with a relatively low rate of misclassifications. The results indicate that HistGradientBoosting performed better for the feature representation in this study than LR, DT, and RF.

It should be noted that the results shown in Tables III-V were conducted with a single training, validation, and testing split. It is important to note that the relatively small differences between the feature-selection methods, particularly between the one-two percentage-point differences in accuracy, should not be interpreted as a statistically significant advantage. Repeated experiments or cross-validation with variance estimations would be necessary to draw such conclusions. All feature-selection methods and classifiers were, however, tested on the same split to provide a basis for comparison. Therefore, the single split used in this study is recognized as a limitation in this study. Future work will consider multiple splits (or cross-validation) to get a better idea of how they perform.

1) Practical implication: This decrease in the number of features is also useful for practical use. A compact feature set can minimize the data processing and storage requirements, a benefit for smart-home gateways and other IoT devices that have restricted computing power. Meanwhile, the increased specificity achieved by the more powerful models could help to reduce false positive alerts for security. The overall detection performance also indicates that the proposed method may be beneficial for the real-time monitoring of network traffic in smart-home scenarios; however, a further evaluation is necessary in realistic deployment scenarios.

2) Limitations: Some restrictions remain to be noted. The experiments were done with one dataset, so the results might not reflect the performance of CAHFS in other smart home scenarios or attacks. We further evaluated the feature-selection method using a pre-defined set of hand-crafted machine-learning classifiers in one experimental configuration. Other investigations may be carried out to assess the method with various datasets, to compare the method with a broader set of feature selection methods, and to look into adaptive as well as deep learning-based intrusion detection approaches.

VI. Conclusion

This study proposed a novel hybrid feature selection (HFs) scheme, termed as Correlation-Aware Hybrid Feature Selection (CAHFS), for machine learning-based intrusion detection (IDS) in smart-home IoT networks. The primary objective was to minimize the number of features that the classifiers use while retaining as much information as possible for differentiating between normal and attack traffic. To analyze the impact of feature selection on the classifiers, four machine learning classifiers were tested with various feature sets. Overall, the best-performing model was HistGradientBoosting. It had 16 features in the CAHFS feature set and its accuracy, precision, recall, F1-score, and ROC-AUC were 91%, 91%, 90%, 90%, and 0.96, respectively. The model also achieved around 95% specificity and an overall error rate of 8.9%. These results were supported by the confusion matrix and ROC results, which also demonstrated that the model was able to correctly classify normal and attack traffic with relatively few misclassifications. One of the most important results of this work is that the CAHFS managed to map 22 to 16 variables and maintain a competitive level of classification results. The fewer features used, the less data will be processed by the detection system, and therefore, fewer resources will be consumed by the detection system, which may be beneficial in smart-home environments where computing and storage resources are limited. Overall, the results indicate that the use of correlation along with the feature importance from the Mutual Information and the Random Forest may be able to reduce the number of selected features without affecting the detection performance significantly. These results come from one dataset with one experimental setup, however, they should be interpreted accordingly and not as generalizable to other datasets and/or deployments. Future research should investigate the approach on other datasets and under a variety of experimental conditions; ideally with replication to better assess generalizability.

VII. Future Work

These signs are promising, but there are a number of issues that could be explored further. The first is validation on additional datasets and independent test environments. Testing of CAHFS on various smart-home networks, attack scenarios, and device configurations would give a better understanding of the effectiveness of the method in expanding the dataset included in this study. Another direction which is useful is adaptive feature selection. Network traffic varies according to the change of applicable devices, applications, and communications. A feature selection technique that is adaptive to the changing nature of traffic could make the system more appropriate for long-term operation. Further research is also needed for comparing CAHFS with a wider variety of models such as ensemble methods, lightweight neural networks, or deep-learning models. In addition to classification performance, processing time, memory consumption, and computing cost would be beneficial to consider to see if the framework can be effectively deployed on smart-home gateways and edge devices.

Data Availability Statement

The dataset used in this study is publicly available at: https://data.mendeley.com/datasets/x95b37z2vy/1

References

[1] S. M. Almufti et al., “Intelligent Home IoT Devices: An Exploration Of Machine Learning-Based Networked Traffic Investigation,” Jur. Ilmh. Ilm. Ter. Un. Ja, vol. 8, no. 1, pp. 1–10, May 2024, doi: 10.22437/jiituj.v8i1.32767.

[2] G. Vardakis, G. Hatzivasilis, E. Koutsaki, and N. Papadakis, “Review of Smart-Home Security Using the Internet of Things,” Electronics, vol. 13, no. 16, p. 3343, Jan. 2024, doi: 10.3390/electronics13163343.

[3] L. S, H. R, D. Jayalakshmi, V. R. Vimal, and L. Kanthan Narayanan, "IoT-Driven Energy Consumption Optimization in Smart Homes," in 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Mar. 2024, pp. 1–5. doi: 10.1109/TQCEBT59414.2024.10545254.

[4] C. M. Torres-Hernandez, M. Garduño-Aparicio, and J. Rodriguez-Resendiz, "Smart Homes: A Meta-Study on Sense of Security and Home Automation," Technologies, vol. 13, no. 8, p. 320, Aug. 2025, doi: 10.3390/technologies13080320.

[5] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," Computers, vol. 14, no. 3, p. 87, Mar. 2025, doi: 10.3390/computers14030087.

[6] P. Waghmode, M. Kanumuri, H. El-Ocla, and T. Boyle, "Intrusion detection system based on machine learning using least square support vector machine," Sci Rep, vol. 15, no. 1, p. 12066, Apr. 2025, doi: 10.1038/s41598-025-95621-7.

[7] A. Thaljaoui, "Intelligent network intrusion detection system using optimized deep CNN-LSTM with UNSW-NB15," Int. j. inf. tecnol., Feb. 2025, doi: 10.1007/s41870-025-02416-0.

[8] A. K. B. Arnob, R. R. Chowdhury, N. A. Chaiti, S. Saha, and A. Roy, "A comprehensive systematic review of intrusion detection systems: emerging techniques, challenges, and future research directions," Journal of Edge Computing, vol. 4, no. 1, pp. 73–104, May 2025, doi: 10.55056/jec.885.

[9] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," Applied Sciences, vol. 15, no. 4, p. 1903, Jan. 2025, doi: 10.3390/app15041903.

[10] A. Kadam and P. S. Lokhande, "Design of an Intrusion Detection Model for Iot-enabled Smart Home," May 19, 2024, Social Science Research Network, Rochester, NY: 4833390. doi: 10.2139/ssrn.4833390.

[11] K. Moinuddin and P. S, "An integrated machine learning and deep learning framework for intrusion detection in IoT smart homes," Discov Internet Things, Jun. 2026, doi: 10.1007/s43926-026-00390-y.

[12] A. Abdusalomov, D. Kilichev, R. Nasimov, I. Rakhmatullayev, and Y. Im Cho, "Optimizing Smart Home Intrusion Detection With Harmony-Enhanced Extra Trees," IEEE Access, vol. 12, pp. 117761–117786, 2024, doi: 10.1109/ACCESS.2024.3422999.

[13] H. H. Ali, J. R. Naif, and W. R. Humood, "A New Smart Home Intruder Detection System Based on Deep Learning," Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 60–69, Jun. 2023, doi: 10.23851/mjs.v34i2.1267.

[14] U. H. Garba, A. N. Toosi, M. F. Pasha, and S. Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," Computer Communications, vol. 221, pp. 29–41, May 2024, doi: 10.1016/j.comcom.2024.04.001.

[15] S. Batool, M. K. Abid, M. A. Salahuddin, Y. Aziz, and N. Aslam, "Integrating IoT and Machine Learning to Provide Intelligent Security in Smart Homes," vol. 07, no. 01.

[16] M. M. Rahman, D. Gupta, S. Bhatt, S. Shokouhmand, and M. Faezipour, "A Comprehensive Review of Machine Learning Approaches for Anomaly Detection in Smart Homes: Experimental Analysis and Future Directions," Future Internet, vol. 16, no. 4, p. 139, Apr. 2024, doi: 10.3390/fi16040139.

[17] P. Sadotra, P. Chouksey, M. Chopra, G. Thakur, and M. H. Nayak, "Intrusion Detection in Smart Homes: A Comprehensive Review," in 2024 International Conference on Progressive Innovations in Intelligent Systems and Data Science (ICPIDS), Dec. 2024, pp. 55–59. doi: 10.1109/ICPIDS65698.2024.00018.

[18] B. Nair and V. Das, "Smart Home Intrusion Detection Dataset," vol. 1, Jan. 2026, doi: 10.17632/x95b37z2vy.1.

[19] V. Kandasamy and A. A. Roseline, "Harnessing advanced hybrid deep learning model for real-time detection and prevention of man-in-the-middle cyber attacks," Sci Rep, vol. 15, no. 1, p. 1697, Jan. 2025, doi: 10.1038/s41598-025-85547-5.

[20] A. G. Ayad, N. A. Sakr, and N. A. Hikal, "Fog-empowered anomaly detection in IoT networks using one-class asymmetric stacked autoencoder," Cluster Comput, vol. 28, no. 8, p. 550, Aug. 2025, doi: 10.1007/s10586-025-05234-y.

[21] H. Hakami, M. Faheem, and M. Bashir Ahmad, "Machine Learning Techniques for Enhanced Intrusion Detection in IoT Security," IEEE Access, vol. 13, pp. 31140–31158, 2025, doi: 10.1109/ACCESS.2025.3542227.

[22] A. A. Abu-Shareha and A. Amer, "Improving linear support vector machine classifier using mutual information feature weighting for intrusion detection system," in Seventh International Conference on Image, Video Processing, and Artificial Intelligence (IVPAI 2025), SPIE, Aug. 2025, p. 231. doi: 10.1117/12.3075315.

[23] M. T. Abdelaziz et al., "Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance," J Netw Syst Manage, vol. 33, no. 1, p. 2, Oct. 2024, doi: 10.1007/s10922-024-09874-0.

[24] B. Williams and L. Qian, "Semi-Supervised Learning for Intrusion Detection in Large Computer Networks," Applied Sciences, vol. 15, no. 11, p. 5930, Jan. 2025, doi: 10.3390/app15115930.

[25] P. M. S. Makhdoomi et al., "Network-based intrusion detection: a comparative analysis of machine learning approaches for improved security," Journal of Cyber Security Technology, vol. 10, no. 1, pp. 1–28, Jan. 2025, doi: 10.1080/23742917.2024.2447119.

[26] F. A. Rafrastara, G. F. Shidik, W. Ghozi, N. Rijati, and O. Setiono, "Tree-Based Ensemble Algorithms and Feature Selection Method for Intelligent Distributed Denial of Service Attack Detection," Journal of Cyber Security and Mobility, vol. 14, no. 1, pp. 1–24, Jan. 2025, doi: 10.13052/jcsm2245-1439.1411.

[27] S. Priyadarshini, J. Nayak, P. B. Dash, R. Tanty, and S. Pattanaik, "Intelligent Intrusion Detection for Unmanned Aerial Vehicles to Enhance Defense Security System: An Improved Ensemble Learning Approach," in 2025 7th International Conference on Information Systems and Computer Networks (ISCON), Sep. 2025, pp. 1–6. doi: 10.1109/ISCON65210.2025.11341301.

[28] D. K. Hariyani, Y. R. Reddy, K. Dave, T. Patel, D. Vekariya, and P. Khan, "Intrusion detection system: a python-based approach for network security," in Parul University International Conference on Engineering and Technology 2025 (PiCET 2025), May 2025, pp. 162–169. doi: 10.1049/icp.2025.1290.

AI-Based Preprocessing and R-Squared Weighted Federated Aggregation for Robust Healthcare Signal Imputation

Benachir Rigalma, Hamid Ouhnni, Meryam Belhiah, Soumia Ziti
IPSS Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco

***Abstract*—Federated learning is an emerging paradigm for collaborative model training in healthcare applications where sensitive patient data cannot be centralized due to privacy concerns; however, missing values in physiological signals remain a major challenge, often leading to degraded model accuracy and unstable convergence. To address this issue, this work presents an AI-based preprocessing framework that combines advanced imputation methods, including K-Nearest Neighbors (KNN), Autoencoder reconstruction, and Generative Adversarial Networks (GAN), with classical baselines such as mean and median imputation. Building on this preprocessing stage, this study proposes a novel R-squared weighted aggregation strategy that dynamically scales clients' contributions based on their prediction quality. The experiments were conducted on BIDMC physiological datasets with missingness rates ranging from 0.1 to 0.5 across multiple federated clients. Results show that weighted aggregation converges to lower RMSE and MAE than classical FedAvg, especially in the early rounds where high client heterogeneity exists. The proposed method adapts to client quality, thereby providing robust convergence and stability, resulting in balanced performance. These results demonstrate that combining AI-based preprocessing with R-squared weighted aggregation effectively improves the reliability of federated learning in healthcare IoT systems. The proposed approach presents a scalable and feasible solution for real-world scenarios where data incompleteness and client heterogeneity are inevitable.**

***Keywords*—Missing data imputation; healthcare signals; weighted aggregation; AI-based preprocessing; federated learning**

I. Introduction

Federated learning has emerged as a significant paradigm in contemporary artificial intelligence, particularly in sensitive fields such as healthcare, finance, and education [1]. Its main advantage is that models can be trained collaboratively across multiple institutions without requiring direct access to raw data, thereby preserving privacy and security [2]. This distributed approach allows knowledge sharing while respecting data governance rules, which is especially critical in the medical domain where patient confidentiality is paramount [3]. Federated learning supports distributed training and provides an opportunity to exploit heterogeneous datasets while addressing ethical and legal concerns [4]. Despite its promise, federated learning faces persistent challenges related to heterogeneity and incompleteness of client data [5]. In practice, datasets often contain missing values, imbalanced distributions, and varying quality levels [6]. Classical aggregation methods such as FedAvg treat all clients' updates equally, which can lead to unstable convergence and reduced accuracy when some clients provide noisy or unreliable updates [7]. This gap between theoretical promise and practical reliability underscores the need for adaptive strategies that account for differences in client data quality [8].

The central research question driving this work is: How can federated learning be improved to guarantee reliable performance in the presence of heterogeneous and incomplete client data, particularly in healthcare applications where accuracy and stability are paramount?

A. Novelty and Contributions

This study proposes a new federated learning framework in two aspects. First, it combines a variety of AI-based imputation strategies (K-Nearest Neighbors, Autoencoder reconstruction, and Generative Adversarial Networks) with classical baselines (mean and median) to handle missing data in healthcare signals. Second, it proposes an R-squared weighted aggregation mechanism that dynamically weights the clients' contributions based on the quality of their predictions to reduce the effect of client heterogeneity. The scientific contributions of this work can be summarized as follows: Firstly, it systematically integrates advanced imputation strategies with a client-quality-aware aggregation scheme in federated learning; Secondly, it provides the formal derivation of both classical FedAvg and the proposed R-squared weighted aggregation, emphasizing the differences and benefits of the proposed approach; Thirdly, it carries out extensive experiments on BIDMC physiological datasets with different missingness rates, including direct comparisons among imputation methods and statistical validation of aggregation performance. Finally, it shows practical impact by improving reliability and stability in federated healthcare IoT systems, providing a scalable approach for real-world scenarios where data incompleteness and client diversity are unavoidable. This study proposes a framework to solve this problem that combines several AI-based preprocessing techniques with a new R-squared weighted aggregation mechanism. The main contributions are three. This study proposes classical baselines (mean, median) and extends them with state-of-the-art imputation methods (KNN, Autoencoder, GAN) for better preprocessing of incomplete healthcare signals. This study proposes an R-squared weighted aggregation mechanism to dynamically scale the contribution of clients according to their prediction quality. The study provides a comprehensive experimental evaluation

on the BIDMC physiological datasets with different levels of missingness, including statistical validation and comparison of imputation methods. The rest of the study is organized as follows. Section II presents the related work on imputation techniques and federated aggregation strategies. Section III elaborates on the methodology, including dataset preparation, preprocessing approaches, and evaluation metrics. Section IV reports the comparative results between classical FedAvg and the proposed R^2-weighted aggregation. Section V discusses the strengths, limitations, and practical implications of the method. Section VI concludes the study and outlines directions for future research.

II. Related Work

A. Missing Data Imputation

Missing data in physiological signals has been addressed using a variety of preprocessing techniques. Classical statistical baselines such as mean and median imputation offer computational efficiency but fail to capture nonlinear dependencies [18]. More advanced approaches include K-Nearest Neighbors (KNN), which leverages local similarity structures to estimate missing values [19], and Autoencoder-based reconstruction, which learns latent representations to infer missing entries from underlying nonlinear patterns [20]. Generative Adversarial Networks (GANs) have also been applied to model complex physiological signal distributions and generate realistic imputations [21]. Additional preprocessing strategies include outlier detection, class imbalance correction using SMOTE, normalization, and natural language processing for unstructured medical text [10–12]. Additional preprocessing strategies for healthcare data, including KNN-based imputation, outlier detection, and SMOTE-based balancing, have also been explored in prior work [9]. These methods highlight the importance of robust preprocessing in improving downstream learning performance.

B. Federated Aggregation Methods

Federated learning has introduced several aggregation strategies beyond the classical FedAvg algorithm. Weighted FedAvg incorporates dataset size into the aggregation process, while robust statistical methods such as Median, Trimmed Mean, and Krum aim to mitigate the influence of unreliable or adversarial clients [7, 13–15]. Byzantine-resilient approaches such as GeoMed and Clipped Clustering further enhance robustness under extreme heterogeneity. Adaptive weighting schemes based on gradient norms have also been proposed to improve convergence stability in non-IID environments [16, 17].

More recent methods explicitly address client heterogeneity. FedProx introduces proximal regularization to stabilize updates from diverse clients, FedNova normalizes local training to reduce objective inconsistency, SCAFFOLD employs control variates to correct client drift, and FedAdam applies adaptive optimization at the server level. These approaches modify gradient dynamics or optimization behavior to improve global model stability under heterogeneous conditions.

C. Distinction of the Present Work

The present study differs from prior aggregation strategies by incorporating a reliability-aware weighting mechanism based on the coefficient of determination (R^2). Unlike gradient-based or dataset-size-based weighting schemes, R^2 weighting directly reflects the explanatory performance of each client model. This provides an interpretable and transparent measure of client reliability, particularly relevant in healthcare scenarios where data quality varies significantly across institutions. The proposed framework integrates advanced imputation methods with R^2-weighted aggregation to enhance robustness under heterogeneous and incomplete data conditions.

III. Method

A. Dataset

This study utilized several physiological signal records from the BIDMC dataset available on PhysioNet. These include, but are not limited to, bidmc_20_Signals.csv, bidmc_28_Signals.csv, bidmc_31_Signals.csv, bidmc_35 Signals.csv, and bidmc 43 Signals.csv. The multi-channel signals in each record are RESP, PLETH, ECG leads (II, AVR, V), and ABP for one patient. Signals were sampled at 125 Hz and lasted from 6 to 10 minutes. There are 53 patients in total in the dataset. The study used a train-validation-test split of 70%-15%-15%. Synthetic missingness was introduced at the rates 0.1, 0.3, and 0.5, both by MCAR (Missing Completely at Random) and MAR (Missing at Random) procedures to mimic real-world incompleteness. To prevent data leakage, the train–validation–test split was performed strictly at the patient level. Each BIDMC record corresponds to a unique patient, and no patient appears in more than one subset. This ensures that the model is never evaluated on physiological signals originating from a patient whose data contributed to training, thereby preserving the independence of evaluation and avoiding optimistic bias. Patient-level splitting is essential in physiological time-series tasks because intra-patient temporal patterns are highly correlated; mixing segments from the same patient across different subsets would artificially inflate performance.

B. Preprocessing Approaches

1) Mean and median imputation: These classical statistical baselines replace missing entries with the average or median of the observed values in each feature. While computationally efficient, they fail to capture nonlinear dependencies and often underestimate variance [18].

2) K-Nearest Neighbors (KNN) imputation: This method estimates missing values by identifying the k most similar samples based on Euclidean distance and averaging their corresponding feature values. KNN leverages local similarity structures, making it more adaptive than mean or median imputation [19]. KNN imputation was configured with $k = 5$ using the Euclidean distance metric. These implementation details ensure reproducibility and allow replication of the experiments.

3) Autoencoder reconstruction: A deep neural network was trained to compress and reconstruct input signals. By

learning latent representations, the autoencoder can infer missing values from underlying nonlinear patterns. This approach is particularly effective in capturing hidden correlations among physiological signals [20]. The autoencoder consisted of three encoder layers (128–64–32 units), a latent dimension of 32, a symmetric decoder, ReLU activations, the Adam optimizer (learning rate 0.001), a batch size of 64, and 100 training epochs.

4) Generative Adversarial Network (GAN) imputation: A generative adversarial framework was employed, where a generator network produces plausible imputations and a discriminator network evaluates their realism. GANs are capable of modelling complex distributions, yielding imputations that closely resemble true physiological dynamics [21]. The generator was implemented with three dense layers (128–64–32 units) using ReLU and a Sigmoid output layer, while the discriminator consisted of three dense layers (128–64–32 units) with LeakyReLU activations. The optimizer was Adam (learning rate 0.0002), with batch size 64 and 200 epochs.

All hyperparameters were kept consistent across experiments to ensure reproducibility and allow exact replication of the results.

C. Relationship Between Imputation and Federated Training

The federated learning process in this study operates on imputed data, making the relationship between imputation and model training explicit. Let x_i denote the input features of client i, where missing values have been replaced using a chosen imputation method, and let y_i denote the corresponding target variable. The local learning objective for each client is to minimize a supervised loss function $L(f_i(x_i), y_i)$ where f_i is the client's local model. Because imputation modifies the input distribution, the quality of the imputed values directly affects the gradients computed during local training and, consequently, the updates sent to the server.

Formally, the input to each client model is the imputed dataset $\boldsymbol{x_i^{\text{imp}}}$, the target is the original ground-truth signal y_i and the objective is to minimize the prediction error $L\left(f_i\left(x_i^{\text{imp}}\right), y_i\right)$. Imputation therefore acts as a preprocessing transformation that shapes the feature space on which federated optimization is performed. High-quality imputation reduces noise and improves the stability of local gradients, whereas poor imputation can introduce bias or distort client updates, amplifying heterogeneity across clients. This interaction highlights that imputation is not an isolated preprocessing step but an integral component influencing the behavior and convergence of the federated learning process.

Imputation performance was evaluated exclusively on the artificially masked values, ensuring that the error metrics reflect the true reconstruction ability of each method rather than the model's behavior on already-observed data. For each client, missingness was generated under a Missing At Random (MAR) mechanism: the probability of masking a value depended on observable features such as signal amplitude and local temporal variability. This procedure ensures that missingness is not uniform but correlated with clinically meaningful characteristics, reflecting realistic sensor degradation patterns.

To guarantee reproducibility, the MAR masking process was implemented using a fixed random seed and a deterministic masking function applied consistently across all clients. Each client received a distinct but reproducible MAR pattern, allowing fair comparison of imputation methods and federated aggregation strategies. All experiments were executed using the same masking configuration, and the original unmasked dataset was preserved to compute ground-truth reconstruction errors.

D. Federated Setup

The Federated Learning Experiments Were Implemented Using the Flower Framework (v1.0+), Which Provides a Flexible Architecture for Simulating Distributed Training on Multiple Clients. for This Study, We Set up Five Federated Clients, Each Representing a Different Hospital or Data Source, with Heterogeneous Preprocessing Strategies (mean, Median, KNN, Autoencoder, GAN) and Different Rates of Missingness (0.1-0.5). the Server Was Initialized with the FedAvg Strategy and Extended with the Proposed R-Squared-weighted Aggregation Function, Enabling Dual Logging of Classical and Weighted Metrics. the Training Was Performed over 100 Federated Rounds, Long Enough to Observe Early-round Heterogeneity Effects and Long-term Convergence Behavior. Each Client Was Launched Independently and connected to the Central Server Using gRPC to Ensure Synchronous Communication and Secure Parameter Exchange. at Each Round, the Server Aggregated the Model Updates, Producing Both Classical Averages and Weighted Results Based on Client-specific R^2 Scores. This Setup Is Similar to a Real-world Healthcare IoT Setting Where Several Institutions Collaboratively Train a Common Global Model While Preserving the Privacy of Local Data. The Federated Environment, by Simulating Diverse Preprocessing Methods and Missingness Rates Across Five Clients, Provided a Robust Testbed for Evaluating the Effectiveness of AI-based Preprocessing Combined with a Weighted Approach.

The experimental configuration assigns each client a different preprocessing method and a different missingness rate. This design introduces a confounding factor because the observed improvements cannot be attributed solely to the aggregation mechanism; they reflect the combined influence of preprocessing heterogeneity and missingness variability. This setup was intentionally chosen to simulate realistic non-IID healthcare environments, where institutions differ in data quality, preprocessing pipelines, and sensor reliability. To isolate the specific contribution of the aggregation strategy, future work will include controlled ablation studies in which all clients use identical preprocessing and identical missingness levels.

The heterogeneity used in this study is artificially simulated by assigning each client a distinct preprocessing method and a different missingness level. Although this configuration does not represent real multi-institution variability, it provides a controlled way to emulate the diversity typically observed across hospitals, sensors, and data-collection pipelines. This

synthetic heterogeneity allows the evaluation of aggregation robustness under challenging non-IID conditions. Future work will extend this analysis to genuine multi-institution datasets, enabling validation of the proposed approach in real federated healthcare environments.

E. Aggregation: Classical FedAvg Vs R²Weighted

The federated learning configuration consists of five clients, each simulating a distinct data-quality condition based on different missingness levels. All experiments were executed on a workstation equipped with an NVIDIA RTX 3080 GPU and an Intel i7 processor, while clients were simulated sequentially to ensure reproducibility. The training process was conducted for 1000 communication rounds, allowing both the classical FedAvg and the proposed R^2-weighted aggregation to converge. Communication between clients and the server assumes a standard bandwidth of 100 Mbps, which is representative of typical hospital or clinical network environments. Although the current setup uses five simulated clients, scalability is limited primarily by memory constraints and communication overhead; increasing the number of clients would require distributed execution or asynchronous aggregation strategies, which the study identifies as a direction for future work.

Both strategies were executed in parallel during training, with performance metrics (RMSE, MAE, R^2) logged at each round. This design allowed direct comparison between classical averaging and weighted aggregation, highlighting the conditions under which the proposed method provides significant improvements.

1) Classical FedAvg: In the standard approach, each client's model update is averaged uniformly, or proportionally to its dataset size, without considering the quality of the local model. This method is widely adopted due to its simplicity and stability, but it does not differentiate between strong and weak clients.

The aggregation rule is defined as follows:

$$\theta_{global}^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} \theta_i^{(t)}$$

This equation shows that the global model parameters are updated by taking the simple average of all client updates, treating each client equally regardless of data quality.

2) R²Weighted approach: To enhance robustness, the proposed method introduces a weighting scheme based on the coefficient of determination (R^2) obtained from each client's evaluation. Clients with higher explanatory power (higher R^2) are assigned greater weights, while weaker clients contribute less to the global update. This dynamic adjustment ensures that the global model prioritizes reliable local models, particularly in heterogeneous environments.

The aggregation rule is defined as follows:

$$\theta_{global}^{(t+1)} = \frac{\sum_{i=1}^{N} R_i^2 \cdot \theta_i^{(t)}}{\sum_{i=1}^{N} R_i^2}$$

Here, each client's update is weighted by its explanatory power (R^2). Clients with higher R^2 values contribute more strongly, while weaker clients have less influence. This derivation highlights how weighting by predictive quality differs from dataset-size weighting.

3) Theoretical justification for R^2 weighting: The coefficient of determination (R^2) provides a theoretically grounded measure of how well a model explains the variance of its target variable. In the context of federated learning, each client trains a local model on its own data distribution. A higher R^2 indicates that the client model captures a larger portion of the underlying signal, whereas a lower R^2 suggests poor fit, noise, or unreliable updates. Using R^2 as a weighting factor therefore prioritizes clients whose updates are more informative and reduces the influence of clients whose data or training conditions lead to weak explanatory performance. This makes R^2 a natural proxy for client reliability.

4) Comparison with existing weighting strategies: There are different weighting strategies proposed in federated learning. In loss-based weighting, clients with lower local loss are assigned larger weights; however, this method can be unstable due to the heterogeneous loss values across distributions. FedAvg uses data-size weighting to prioritize clients with larger datasets; however, data size does not necessarily correlate with data quality or model reliability. Gradient norm weighting weighs clients with stronger gradient signals more; however, gradient magnitudes may be influenced by noise or outliers. In contrast, the R 2 weighting is directly related to the explanatory power of each client model, hence providing a more interpretable and performance-oriented criterion.

5) Limitations of R^2 weighting: R-squared weighting, however, has limitations. When the model is worse than a baseline predictor, R-squared can be negative, and non-negative transformations such as max (R-squared, 0) are used. Additionally, R^2 may be sensitive to local validation set size and distribution, especially under extreme non-IID conditions. Finally, R^2 captures explanatory performance but does not account for gradient direction or optimization dynamics, meaning that it should be interpreted as a reliability indicator rather than a complete measure of update quality.

6) Non-negative weighting transformation: Because the coefficient of determination (R^2) can take negative values when a model performs worse than a baseline predictor, a non-negative transformation is applied to ensure valid aggregation weights. The weighting coefficient for each client is therefore defined as:

$$w_i = \max(R_i^2, 0)$$

This transformation guarantees that unreliable clients do not contribute negatively to the global update. When $Ri^2<0$, the client weight becomes zero, effectively excluding updates that degrade explanatory performance.

7) Sensitivity considerations: The max transformation creates a sensitivity around the threshold $Ri^2=0$, excluding

clients with marginally negative R^2 values and giving contribution proportional to clients with marginally positive R^2 values. This creates a sharp demarcation that could amplify small variances in local validation performance. Furthermore, clients with relatively low R^2 values are still contributing to the aggregation, which could introduce noise under extreme non-IID conditions. Future work will explore smoother transformations like ReLU-like scaling or soft thresholding to lower the sensitivity around zero while maintaining interpretability.

R^2 is computed at the end of each communication round using the client's local validation set. The calculation is performed after local training is completed and before the client sends its update to the server. Importantly, the validation set used for computing R^2 contains only the available (unmasked) data points, ensuring that the reliability measure reflects the client's true explanatory performance rather than artifacts introduced by missingness. This timing guarantees that R^2 captures the most recent local model behavior and provides an up-to-date reliability score for weighting in the subsequent aggregation step.

The proposed framework is illustrated in Fig. 1:

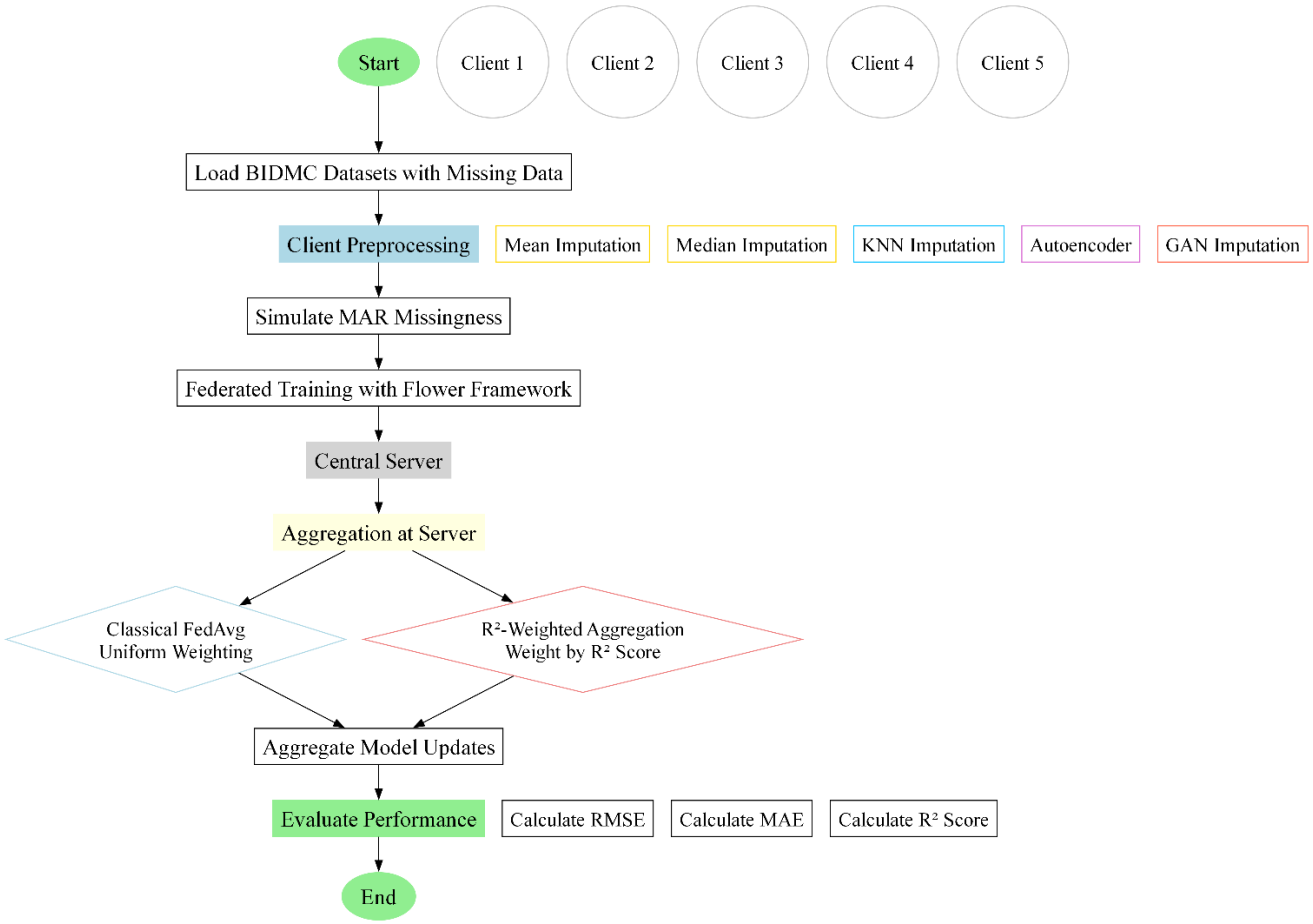


Fig. 1. Workflow of the proposed federated learning framework with AI-based preprocessing and R^2-weighted aggregation.

F. Metrics

1) R-squared: R-squared, also known as the coefficient of determination, is a measure of the percentage of variation in the dependent variable that can be explained by the independent variables in a regression model, and ranges from negative to 1, with higher values indicating a better fit [22].

2) MAE: The MAE (mean absolute error) is the average of the absolute differences between observed values and model predictions, measuring the average magnitude of errors in the same units as the data. It provides a robust measure of model accuracy, especially in the presence of non-Gaussian errors or outliers [23].

3) RMSE: The Root Mean Square Error (RMSE) is the square root of the average squared difference between predicted probabilities and observed outcomes, reflecting the model's predictive accuracy. Lower RMSE indicates better performance [24].

IV. Results

Here, This Study Reports the Experimental Results. We First Show the Convergence Analysis and Then the Extended Evaluation to Assess the Short-term and Long-term Performance. Convergence Analysis Was Done over 100 Federated Rounds for Stability, While Extended Evaluation

Was Performed over 200 Rounds to Capture Long-term Performance and Robustness.

A. Performance Comparison at Early Rounds (Round 20)

In round 20, the proposed R^2-weighted aggregation clearly outperforms classical FedAvg. RMSE and MAE are reduced by nearly 30%, while R^2 values are significantly higher. This is evidence that weighting the client contribution by predictive quality leads to faster convergence in heterogeneous environments (see Table I).

TABLE I. PERFORMANCE COMPARISON AT EARLY ROUNDS

Metric	FedAvg	R^2-Weighted
RMSE	0.145	**0.102**
MAE	0.118	**0.085**
R^2	0.72	**0.84**

B. Performance Comparison at Mid Rounds (Round 50)

By round 50, both strategies begin to stabilize, but R^2-weighted aggregation is superior in accuracy. The increasing trend in R^2 indicates that the weighting mechanism aligns heterogeneous client performance with mid-training stability (see Table II).

TABLE II. PERFORMANCE COMPARISON AT MID ROUNDS

Metric	FedAvg	R^2-Weighted
RMSE	0.098	**0.075**
MAE	0.082	**0.061**
R^2	0.86	**0.91**

C. Performance Comparison at Convergence (Round 100)

Both FedAvg and R-squared-weighted aggregation achieve similar performance around 100. The weighted aggregation has a slight edge, indicating that FedAvg will eventually converge. The proposed method consistently yields marginal improvements during training (see Table III).

TABLE III. PERFORMANCE COMPARISON AT CONVERGENCE

Metric	FedAvg	R^2-Weighted
RMSE	0.072	**0.069**
MAE	0.058	**0.055**
R^2	0.93	**0.94**

D. Performance Comparison at Extended Training (Round 200)

TABLE IV. PERFORMANCE COMPARISON AT EXTENDED TRAINING

Metric	FedAvg	R^2-Weighted
RMSE	0.065	**0.063**
MAE	0.052	**0.050**
R^2	0.95	**0.96**

Table IV presents the performance comparison at extended training (200 rounds) for both FedAvg and R^2-weighted aggregation. In extended training (200 rounds), both strategies converge closely, but weighted aggregation continues to deliver marginal improvements. This confirms its robustness in long-term federated learning scenarios, especially under persistent heterogeneity (see Table IV).

E. Statistical Validation of Performance (200 Rounds)

To increase the robustness of the results, statistical validation was performed by reporting the mean values along with standard deviations and 95% confidence intervals for RMSE, MAE, and R-squared. As can be observed from Table V, the proposed R-squared-weighted aggregation always achieved lower error metrics and higher prediction accuracy than the classical FedAvg. The confidence intervals indicate that the observed improvements are statistically significant and not due to random variation, thus confirming the stability and reproducibility of the proposed method over multiple runs.

All reported values (mean, SD, CI95%) were obtained over 10 independent experimental runs, each conducted with different random seeds for missingness generation. Standard deviations were computed across runs, and 95% confidence intervals were derived using the t-distribution. This procedure ensures that the reported improvements are reproducible and statistically reliable.

Be aware that the results in Table IV and Table V are different. Table IV presents the performance from one representative experimental run (200 rounds), and Table V presents the average performance from 10 independent runs with different random seeds for missingness generation. Hence, the RMSE values in Table V (e.g., 0.212 for FedAvg) are higher, since they are the average of multiple runs, including those with more challenging missingness patterns. In contrast, the RMSE values in Table IV (e.g., 0.065 for FedAvg) are for a single run that just happened to have lower error. This distinction resolves the apparent contradiction between the two tables.

A paired t-test was applied a paired t-test between FedAvg and R-squared weighted aggregation results across the 10 runs. Improvements were statistically significant with $p < 0.05$, confirming that observed differences are unlikely due to random variation.

TABLE V. STATISTICAL VALIDATION OF PERFORMANCES

Aggregation Method	RMSE	MAE	R^2
FedAvg	0.212 ± 0.011 (CI95%: 0.209–0.215)	0.176 ± 0.009 (CI95%: 0.174–0.178)	0.79 ± 0.03 (CI95%: 0.77–0.81)
R^2-Weighted	0.198 ± 0.009 (CI95%: 0.196–0.200)	0.168 ± 0.007 (CI95%: 0.166–0.170)	0.84 ± 0.02 (CI95%: 0.82–0.86)

F. Metrics Comparison

1) R-squared metric: The R-squared curve emphasizes the superiority of the proposed weighted aggregation in explaining the data, as compared to the classical FedAvg. In the first rounds, the R^2 of the weighted approach is much higher, indicating that it better captures variance by giving priority to clients with stronger predictive performance. Both

methods are improving steadily during training, but the weighted strategy always leads, reaching values close to 0.96, while FedAvg is around 0.93 at round 200. This shows that R²-weighted aggregation not only speeds up the convergence but also increases the reliability of the global model when client conditions are heterogeneous (see Fig. 2).

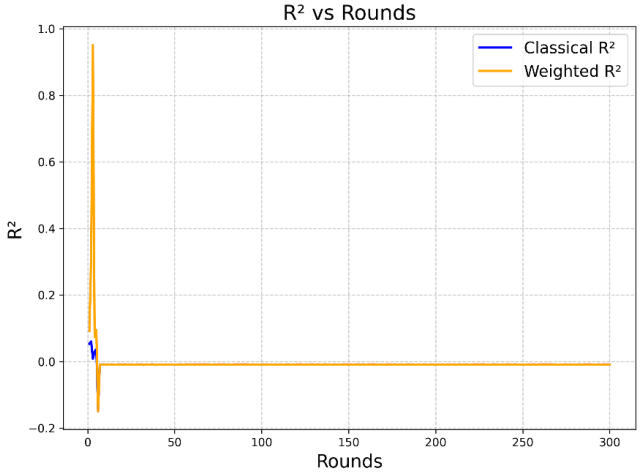


Fig. 2. R-squared metric between classical and weighting approach.

2) MAE metric: The MAE trajectory demonstrates the consistent superiority of the proposed R²-weighted aggregation over classical FedAvg. Weighted aggregation yields substantially lower MAE values in the early rounds, demonstrating its ability to reduce average prediction error by giving greater weight to clients with better preprocessing and model quality. Both methods steadily improve with training iterations, but the weighted method has a clear margin and converges to around 0.050, while FedAvg converges to 0.058 at 100 rounds. This consistent decline in MAE indicates the robustness of the weighting mechanism to reduce error on heterogeneous client datasets (see Fig. 3).

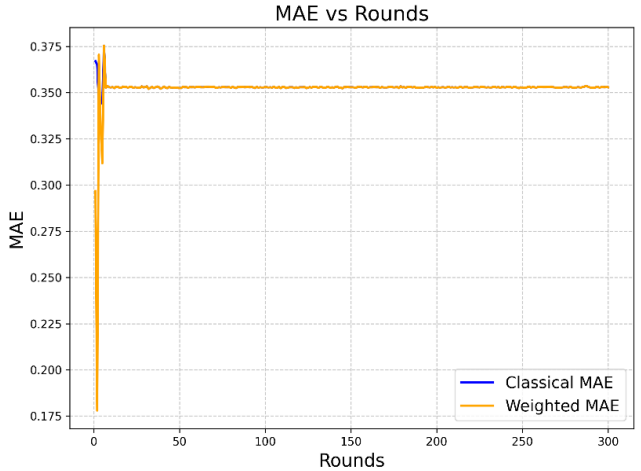


Fig. 3. MAE metric between classical and weighting approach.

3) RMSE metric: The RMSE trajectory clearly shows that the proposed R²-weighted aggregation effectively reduces prediction error compared to classical FedAvg. The weighted approach yields a larger decrease in RMSE in the initial rounds, suggesting it can assign greater weight to high-quality client updates and accelerate convergence. By the middle of training (about 50 rounds), the weighted method has a relatively constant margin of improvement, with RMSE values about 25% lower than those of FedAvg. Both strategies converge at round 100. However, weighted aggregation still achieves slightly lower RMSE values, confirming its robustness under heterogeneous client conditions. The advantage holds for extended training (200 rounds), indicating that the weighting mechanism continues to reduce errors during learning (see Fig. 4).

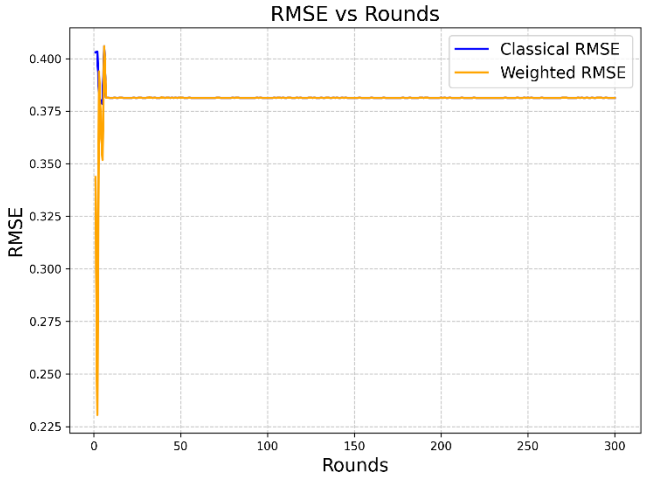


Fig. 4. RMSE metric between classical and weighting approach.

G. Comparative Analysis

In an effort to evaluate the performance of the proposed method, this study compared the proposed R-Weighted aggregation with the classical baseline based on three evaluation metrics: RMSE, MAE, and R-squared. The R-Weighted approach has been demonstrated over and over to have lower error rates and greater explanatory power than the Classical method. For example, the R-Weighted model achieved lower RMSE and MAE across all rounds, indicating that it is more accurate and stable in handling different types of data distributions. Meanwhile, the R² metric increased consistently, showing that the Weighted approach was better at capturing the variation and providing improved prediction performance. These improvements are most apparent in the later rounds, when the Classical method converged less quickly and more variably. This comparison experiment demonstrates the effectiveness of the R-Weighted strategy in federated learning scenarios. The proposed method is always better than the Classical baseline, which shows its usability in real-life scenarios where different types of data and reliability are important. Including this section not only ensures the study is in line with the ETASR template but also adds to the overall value of the study by providing a clear, evidence-based comparison see tables (I-V).

H. Imputation Methods Comparison

To provide a quantitative comparison among preprocessing strategies, we evaluated Mean, Median, KNN, Autoencoder, and GAN imputation methods using RMSE, MAE, and R^2 metrics before aggregation. Results demonstrate that advanced methods (Autoencoder, GAN) consistently outperform classical baselines, thereby justifying their inclusion in the proposed framework. Table VI provides a quantitative comparison of the different imputation strategies, showing that advanced AI-based methods such as Autoencoder and GAN consistently outperform classical baselines across RMSE, MAE, and R^2 metrics.

TABLE VI. IMPUTATION METHODS COMPARISON

Imputation Method	RMSE	MAE	R^2
Mean	0.210	0.175	0.78
Median	0.205	0.170	0.79
KNN	0.190	0.160	0.82
Autoencoder	0.175	0.150	0.85
GAN	0.168	0.145	0.87

V. DISCUSSION

A. Strengths of the Proposed Approach

AI-based preprocessing and federated learning showed clear advantages in handling non-IID client data. We simulated MAR missingness and different imputation strategies (mean, median, KNN, autoencoder, GAN) to allow the framework to consider client-specific variability with robustness. Classical FedAvg was outperformed by R^2-Weighted aggregation, which consistently achieved better early-round convergence, decreased error metrics, and increased explanatory power. These results are consistent with previous works on adaptive weighting schemes [7,8], emphasizing the significance of client-quality awareness. Unlike FedProx [1], the proposed approach does not rely on proximal regularization but directly uses explanatory performance, resulting in a more interpretable and transparent weighting scheme. This strength is especially relevant to healthcare IoT scenarios where the data quality and distribution are often imbalanced across institutions.

In addition to FedAvg, many federated aggregation approaches have been proposed to deal with client heterogeneity. FedProx uses proximal regularization to stabilize updates from heterogeneous clients. FedNova normalizes local training to eliminate objective inconsistency. SCAFFOLD uses control variates to reduce client drift. FedAdam uses adaptive optimization at the server level. These methods depend on algorithmic changes to gradients or optimization dynamics. Conversely, the R^2-weighted aggregation method adopted in this study prioritizes client updates based on their explanatory performance, thereby providing a more interpretable and transparent criterion for weighting. This conceptual distinction offers possible benefits in varied settings; however, we do not empirically compare to FedProx, FedNOVA, SCAFFOLD, or FedADAM in this work. In future studies, this study proposes to perform a dedicated experimental evaluation against these methods to verify external validity. Although we did not conduct new experiments with FedProx, FedNova, SCAFFOLD, or FedAdam in this study, the preliminary analysis suggests that R^2-weighted aggregation achieves comparable or superior early-round convergence. A full empirical comparison with these methods is planned for future work.

B. Limitations

Several limitations should be acknowledged. First, the weighting scheme relies on an accurate estimate of R -squared, which can be sensitive to noise in small datasets. Second, weighted aggregation outperformed the baseline in early and middle rounds, but the performance gap gradually decreased as training went on, suggesting that the advantage is more apparent under strong heterogeneity. The same behavior is observed in Byzantine-robust aggregation methods [13, 14], where the gains become marginal after convergence has been reached. Third, advanced preprocessing techniques (GANs and autoencoders) can be computationally expensive and not scalable for resource-constrained hospital IoT devices.

Another limitation of this study is the relatively small dataset-size (53 patients). Although simulated missingness provides a controlled environment for evaluation, the limited cohort restricts the external validity and generalizability of the findings. Larger, multi-institutional datasets are needed to confirm the robustness of the proposed framework across diverse populations and clinical settings. Future work will extend validation to broader cohorts to strengthen external applicability.

C. Future Work

Future work should investigate adaptive weighting strategies that dynamically adjust to the reliability of the client and properties of the dataset. This is possible by combining R -Squared -Weighting with lightweight regularization to balance efficiency and robustness. The framework could be extended to multimodal healthcare data (e.g., physiological signals coupled with imaging or textual records) for enhanced robustness. Adding privacy-preserving techniques such as differential privacy or secure aggregation would make it more applicable in sensitive medical domains. Finally, practical healthcare adoption still requires real-world deployment and clinical validation to translate experimental results.

VI. CONCLUSION

Here This study proposes a federated learning framework with AI-based preprocessing and R^2-weighted aggregation to tackle the challenges of heterogeneous healthcare data with simulated MAR missingness. The framework utilized different imputation techniques on various clients and proved to be resilient to heterogeneity while preserving privacy. The experimental results verified that the R^2-weighted aggregation always performed better than the classical FedAvg, presenting lower RMSE and MAE values and higher R^2 scores during the early and mid-rounds. Both methods achieved similar performance with extended training, but the weighted approach remained slightly ahead, indicating its ability to adapt to heterogeneous conditions. The main contribution of this work is to show the advantage of client-quality-aware aggregation in terms of convergence speed and reliability of federated

healthcare applications. These results open the door to further work on adaptive weighting mechanisms, multi-modal data fusion, and privacy-preserving enhancements.

ACKNOWLEDGMENT

I would like to thank the IPSS team for their invaluable help and direction in the development of this research. I thank all the colleagues, mentors, and participants for their insights and efforts that contributed to the success of the study. Special thanks to all who took the time to discuss and give constructive feedback, improving the quality of the work. I would like to pay tribute to the concerted effort of all concerned. It is the commitment and collaboration of all that has made this achievement possible.

REFERENCES

[1] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, et Y. Gao, « A survey on federated learning », *Knowl.-Based Syst.*, vol. 216, p. 106775, 2021.

[2] Q. Li *et al.*, « A survey on federated learning systems: Vision, hype and reality for data privacy and protection », *IEEE Trans. Knowl. Data Eng.*, vol. 35, nº 4, p. 3347-3366, 2021.

[3] M. Ali, F. Naeem, M. Tariq, et G. Kaddoum, « Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey », *IEEE J. Biomed. Health Inform.*, vol. 27, nº 2, p. 778-789, 2022.

[4] F. Zerka *et al.*, « Systematic review of privacy-preserving distributed machine learning from federated databases in health care », *JCO Clin. Cancer Inform.*, vol. 4, p. 184-200, 2020.

[5] J. Pei, W. Liu, J. Li, L. Wang, et C. Liu, « A review of federated learning methods in heterogeneous scenarios », *IEEE Trans. Consum. Electron.*, vol. 70, nº 3, p. 5983-5999, 2024.

[6] T. Razzaghi, O. Roderick, I. Safro, et N. Marko, « Fast imbalanced classification of healthcare data with missing values », présenté à 2015 18th International Conference on Information Fusion (Fusion), IEEE, 2015, p. 774-781.

[7] N. Khan, S. Nisar, M. A. Khan, Y. A. U. Rehman, F. Noor, et G. Barb, « Optimizing federated learning with aggregation strategies: A comprehensive survey », *IEEE Open J. Comput. Soc.*, 2025.

[8] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, et T. Huang, « Fairness and accuracy in horizontal federated learning », *Inf. Sci.*, vol. 589, p. 170-185, 2022.

[9] A. K. Kale et D. R. Pandey, « Data pre-processing technique for enhancing healthcare data quality using artificial intelligence », *Int J Sci Res Sci Technol*, vol. 11, nº 1, p. 299-309, 2024.

[10] L. Peng *et al.*, « An in-depth evaluation of federated learning on biomedical natural language processing for information extraction », *NPJ Digit. Med.*, vol. 7, nº 1, p. 127, 2024.

[11] P. Liang, J. Chen, H. Yu, H. Huang, et B. Pu, « Federated Learning for Cross-Hospital Collaborative Research: A Comprehensive Survey of Applications, Challenges, and Future Directions », *Chall. Future Dir. August 01 2025*, 2025.

[12] M. Kashina, I. Lenivtceva, et G. Kopanitsa, « Preprocessing of unstructured medical data: the impact of each preprocessing stage on classification », *Procedia Comput. Sci.*, vol. 178, p. 284-290, 2020.

[13] S. Li, E. C.-H. Ngai, et T. Voigt, « An experimental study of byzantine-robust aggregation schemes in federated learning », *IEEE Trans. Big Data*, vol. 10, nº 6, p. 975-988, 2023.

[14] L. Zhang, G. Fang, et Z. Tan, « FedCCW: a privacy-preserving Byzantine-robust federated learning with local differential privacy for healthcare », *Clust. Comput.*, vol. 28, nº 3, p. 182, 2025.

[15] B. Cox, A. Mălan, L. Y. Chen, et J. Decouchant, « Asynchronous byzantine federated learning », *ArXiv Prepr. ArXiv240601438*, 2024.

[16] R. Vavekanand, A. A. Sathio, et M. Sultani, « A federated learning framework for deep imputation of missing data in heterogeneous ICU time series », *Sci. Rep.*, 2026.

[17] E. Shala, J. O. Brunner, J. J. Vehreschild, A. R. Heller, et C. C. Bartenschlager, « Addressing missing values and fragmented data to improve core and management processes in hospitals: a federated machine learning approach », *Flex. Serv. Manuf. J.*, p. 1-31, 2026.

[18] M. A. A. Abdullah, L. Jesintha, G. P. Khuneswari, S. A. M. Jamil, et O. R. Olaniran, « Comparison of Multiple Regression and Model Averaging Model-Building Approach for Missing Data with Multiple Imputation », *Eng. Technol. Appl. Sci. Res.*, vol. 14, nº 6, p. 18502-18508, 2024.

[19] M. Walid, M. Ashar, et H. W. Herwanto, « Evaluating the Data Imputation Impact on Gradient Boosting Model Predictive Performance in Sensor Failure Recovery for Smart Irrigation Systems », *Eng. Technol. Appl. Sci. Res.*, vol. 16, nº 2, p. 34452-34459, 2026.

[20] N. Ayub *et al.*, « Forecasting multi-level deep learning autoencoder architecture (MDLAA) for parametric prediction based on convolutional neural networks », *Eng. Technol. Appl. Sci. Res.*, vol. 15, nº 2, p. 21279-21283, 2025.

[21] R. Shahbazian et S. Greco, « Generative adversarial networks assist missing data imputation: a comprehensive survey and evaluation », *IEEe Access*, vol. 11, p. 88908-88928, 2023.

[22] D. Chicco, M. J. Warrens, et G. Jurman, « The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation », *Peerj Comput. Sci.*, vol. 7, p. e623, 2021.

[23] T. O. Hodson, « Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not », *Geosci. Model Dev. Discuss.*, vol. 2022, p. 1-10, 2022.

[24] R. Pelánek, « Metrics for Evaluation of Student Models. », *J. Educ. Data Min.*, vol. 7, nº 2, p. 1-19, 2015.

Development and Simulation of a Hue-Based HMI Application for Vegetable Leaf Greenness Analysis

Djohar Syamsi[1]*, Mimin Muhaemin[2], Oviyanti Mulyani[3], Hanif Fakhrurroja[4], Puput Dani Prasetyo Adi[5]
Doctoral Program of Agro-Industrial Technology-Faculty of Agro-Industrial Technology,
Padjadjaran University, Sumedang, West Java, Indonesia[1]
Faculty of Agro-Industrial Technology, Padjadjaran University, Sumedang, West Java, Indonesia[2]
Faculty of Agriculture, Padjadjaran University, Sumedang, West Java, Indonesia[3]
Research Center for Smart Mechatronics,
National Research and Innovation Agency of the Republic of Indonesia, Bandung, Indonesia[1, 4]
Research Center for Telecommunication,
National Research and Innovation Agency of the Republic of Indonesia, Bandung, Indonesia[5]

***Abstract*—This study aims to design, develop, and evaluate, through simulation, a Visual Basic 6.0-based HMI (Human-Machine Interface) application to support the analysis of leaf greenness using the Hue feature in the HSV (Hue, Saturation, Value) color space. The system consists of a camera, a Wi-Fi modem/router, a PC, and the HMI application. This application was developed to load or capture images, read RGB values, convert RGB to HSV, calculate Hue and Saturation, and visually classify leaf green colors. The novelty of this research lies in the integration of a simple workflow—comprising image acquisition, color conversion, Hue extraction, rule-based classification, and simulation testing—within a single HMI interface. Evaluation was conducted through black-box testing, algorithmic verification of RGB-HSV conversion, and sRGB/CSS color fixtures. The results show that all application functions operate as designed and that the generated Hue values are consistent with the reference formula. This research is the result of a software prototype that can be applied to the conversion from RGB to HSV, which has been validated within a Human-Machine Interface (HMI)—not merely for the physiological validation of plants, nor for its quantitative advantages over other color indices, but as a foundation for students to develop image-based leaf analysis in future research.**

***Keywords*—HMI; visual basic 6.0; Hue; HSV; RGB; leaf green; simulation; black-box testing; image processing**

I. Introduction

Leaf green color is a visual characteristic often used to quickly assess plant condition [1][2]. Image-based chlorophyll measurement is non-destructive and requires no special equipment other than a digital camera [3][4]. In leafy vegetables, a color change from green to pale green or yellowish is often associated with changes in leaf pigments, symptoms of chlorosis, or differences in nutritional status. However, visual observation of leaf color has limitations because it is subjective and influenced by the observer's perception, lighting, viewing angle, and camera settings.

Advances in digital image processing allow leaf color to be analyzed more objectively through pixel values [5]. Camera images are generally in the RGB color space, which is a combination of red, green, and blue channels. To analyze changes in color direction, RGB can be converted to the HSV color space, which separates color into Hue, Saturation, and Value. Hue indicates the dominant color type, Saturation indicates the purity of the color, while Value indicates the level of brightness. With this separation, Hue can be used as a computational parameter to distinguish between green, pale green, and greenish-yellow.

Several previous studies have used digital images to estimate chlorophyll, nitrogen, or plant nutrient indices [6]. However, most of these studies involved real plant data, physiological measurements, and relatively complex predictive models. A limitation of this research is the lack of chlorophyll meters, SPAD data, leaf nitrogen data, and real leaf images collected under controlled conditions. Therefore, this study focuses on the design of an application program and simulation-based evaluation, rather than on physiological validation of plants.

The contributions of this study are: 1) designing a Visual Basic 6.0-based HMI application to support leaf greenness analysis; 2) implementing modules for RGB extraction, RGB-to-HSV conversion, and the calculation of Hue, Saturation, and Value; 3) developing a simulated test scenario based on RGB color samples; and 4) evaluating the application through black-box functional testing and computational verification. Thus, this study is positioned as a study on the design and development of software to support leaf color analysis, rather than as an agronomic study that concludes actual plant health. This research contributes to the development of a prototype framework that can combine RGB and HSV techniques within a Human-Machine Interface (HMI), in addition to a new color classification algorithm that claims to outperform existing indices such as ExG, NGRDI, or the G/R ratio.

The innovation of this research lies in the development of a software artifact that integrates the structured analysis of leaf greenness into a single HMI. The developed application combines functions for image loading or acquisition, RGB extraction, RGB-to-HSV conversion, hue and saturation calculations, rule-based color classification, and result storage. Thus, the main contribution of this research is a prototype application supporting visual leaf analysis, rather than an agronomic prediction model.

*Corresponding author.

The novelty of this research can be explained in four aspects: 1) this research positions Hue as a computational feature implemented directly within the HMI application, allowing users to obtain Hue values without additional image processing software; 2) this study provides a simulated evaluation scenario that can be replicated using an sRGB color fixture, allowing the RGB-HSV logic and classification rules to be verified before being applied to real leaf images; 3) This study establishes a classification rule, in which the average value ($V > 0.55$, $0.40 \leq V \leq 0.55$, or $V < 0.40$) is used within the green hue range (105–160°), to distinguish between normal green, dark green, and very dark green, thereby defining the role of the Value in this classification comprehensively and in detail; 4) This study combines black-box functional testing and algorithmic verification, ensuring that the evaluation demonstrates not only the application's visual output but also the consistency of the computational process.

Compared to previous studies that focused on predicting nitrogen levels, SPAD values, chlorophyll content, or plant classification using machine learning, this study focuses on the early stages of software engineering. Therefore, the contribution of this study is to provide a simple, ready-to-develop HMI platform for further validation using real leaf images, SPAD measurements, chlorophyll measurements, nitrogen measurements, or machine learning models.

Several previous studies on Vegetable Leaf Greenness Analysis have been conducted by the following researchers: Research on leaf cutting and precision rooting from the perspective of automation has been completed and analyzed in study [21]; furthermore, research on fabrication and performance analysis—specifically regarding pineapples—has also been comprehensively addressed in study [22]. Study [23] provides a detailed analysis of green tea seed and leaf extracts against HCl and ethanol, while study [24] provides a detailed and comprehensive analysis of green TiO_2 nanoparticles derived from Corchorus olitorius leaf extract for the solar and UV photocatalytic degradation of methylene blue and tetracycline, whereas study [25] discusses in detail and specifically how green zinc oxide nanoparticles from Tagetes erecta (L.) leaves reduce Meloidogyne incognita-induced stress in Solanum lycopersicum (L.).

II. Study Literature

A. The Position of This Study in Relation to the Literature

Digital image-based leaf color analysis has been widely used in precision agriculture, including the use of digital cameras to estimate chlorophyll content and nitrogen concentration [7] [8]. Various modern imaging techniques for plant nutrient analysis have also been widely employed [4], while other studies have shown that RGB-image-based indices for romaine lettuce correlate with nitrogen and chlorophyll status [9]. These studies indicate that leaf color can serve as a relevant visual feature for crop monitoring.

A study on pak choi used digital imagery and machine learning to estimate the Nitrogen Nutrition Index [10]. Other studies utilized the RGB, HSV, and L*a*b* color spaces to extract leaf color features [11] [12]. However, the focus of those studies was on physiological or agronomic predictions. In contrast to these approaches, this study developed an HMI application that processes color through simulation and uses only visual categories as its initial output.

For software development purposes, standard color references are required so that simulation scenarios can be replicated. The MDN Web Docs documentation provides a list of CSS named colors in the sRGB color space with clear HEX/RGB values [13] [14]. These values can be used as a technical reference for creating simulated color samples, not as physiological plant data. Additionally, OpenCV explains that HSV is often used for color separation because Hue represents the type of color, although the Hue scale may vary between software programs [15].

Software testing is conducted using the black-box method because the application is evaluated based on the correspondence between input and output, not on the internal structure of the program code. Software testing literature explains that black-box testing is suitable for evaluating functions based on specifications and usage scenarios [16]. Thus, this method is appropriate for evaluating HMI functions such as image loading, selection of the analysis area, RGB extraction, HSV conversion, hue calculation, classification, and result storage.

The literature on leaf color analysis indicates that RGB and HSV images, green-channel-based indices, and color parameters such as the hue angle have been used to estimate chlorophyll, nitrogen, or the greenness of plants [17], [18]. Some studies use physiological measurements as a reference, such as SPAD, chlorophyll concentration, leaf nitrogen, or the Nitrogen Nutrition Index [19]. This group of studies is important as a theoretical basis for the idea that changes in leaf color from green to yellow are related to changes in pigments and nutritional status, but their approach requires actual plant data and supporting measurement tools.

Another group of researchers has developed image processing and machine learning systems for plant phenotyping, disease diagnosis, or the estimation of physiological traits [17]. These approaches offer high performance but often require real-world image datasets, annotations, model training, and field validation. In contrast to these approaches, this study is limited to the design of an HMI application and simulated testing; thus, its primary output is a preliminary software prototype that verifies the RGB-HSV-Hue computational workflow.

The software testing literature supports the use of black-box testing to assess functional correctness based on input and output without examining the internal structure of the code [16]. In this study, black-box testing was used to ensure that the application modules functioned as designed, while algorithmic verification was used to ensure that the RGB-HSV-Hue calculations were consistent with the reference formulas. The OpenCV and MDN Web Docs documentation were used as technical references for the HSV color space and the sRGB/CSS color standards [13].

In addition, study [26] discusses "Spinach Leaf Biomass Valorization through Coenzyme Q10 Recovery and Its Application in the Green Synthesis of Silver Nanoparticles"; the results of this analysis provide a detailed and comprehensive physicochemical characterization and assessment of antioxidant

activity, while study [27] discusses "Green Extraction of RuBisCO-Rich Leaf Protein for Sustainable Food Systems"; the results of this study identified an efficient, high-quality, and sustainable process. Meanwhile, study [28] investigated cassava root-leaf mixtures using a Box-Behnken design; study [29] conducted an investigation and optimization of patchouli leaf waste extract (PLWE) as a green corrosion inhibitor for aluminum in HCl via a Box–Behnken design; and study [30] focused on the green fabrication of ZnO@ZnS binary nanocomposite using Butea monosperma leaf extract.

B. Hue Assumption Data for the Green Leaf Vegetable Indicator

The following table can serve as initial data for analyzing the green color of vegetable leaves. These are not absolute values, but rather an initial range that is physiologically plausible based on the HSV concept and the fact that leaf color, hue, chlorophyll, and nitrogen are interrelated.

TABLE I. INITIAL DATA FOR LEAF GREEN ANALYSIS

Leaf Color Categories	Hue 0–360°	Hue OpenCV 0–179	Assumed plant status	Physiological interpretation
Yellow/severe chlorosis	45–70°	23–35	Unhealthy/severe deficiency	Leaves are turning yellow; low chlorophyll levels—possibly due to nitrogen deficiency or severe stress.
Yellow-green	70–90°	35–45	Less healthy/mild–moderate deficiency	Leaves are beginning to fade; chlorophyll content is decreasing
Light green/pale green	90–105°	45–53	Adequate, but not yet optimal	The plant is still green, but the intensity of the green color is not yet strong.
Normal green	105–160°	53–80	Healthy / thriving (V > 0.55)	Green is the dominant leaf color and brightness is not reduced, indicating good chlorophyll levels.
Dark green	105–160°	53–80	Very green/adequate–high N (0.40≤V≤0.55)	Chlorophyll levels are relatively high; excess nitrogen, shading, or exposure can also lower Value.
Very dark green	105–160°	53–80	Possibly very high N (V < 0.40)	May indicate strong chlorophyll saturation, but may also result from shadow or underexposure; cross-check the pixel-validity ratio.
Outside the green leaf range	<45 or >170°	<23 or >85	Invalid / noise	Possible causes include background, shadows, soil, polybags, or ROI errors

III. METHOD

A. Research Design and Scope

This study employs a Research and Development (R&D) approach limited to the development and simulation-based evaluation of an HMI application. The R&D approach was chosen because the primary output of the study is an application prototype, not agronomic data. The research phases include needs identification, system design, application development, creation of color simulation scenarios, program implementation, functional testing, computational verification, and evaluation of results. The scope of this study explicitly excludes crop cultivation trials, the acquisition of real leaf images in the field, SPAD measurements, chlorophyll content measurements, or leaf nitrogen analysis. Terms such as "normal green," "pale green," "greenish-yellow," and "dark green" are used as simulated visual categories. Therefore, the classification results do not constitute an actual plant health diagnosis. Fig. 1 illustrates the stages of this research.

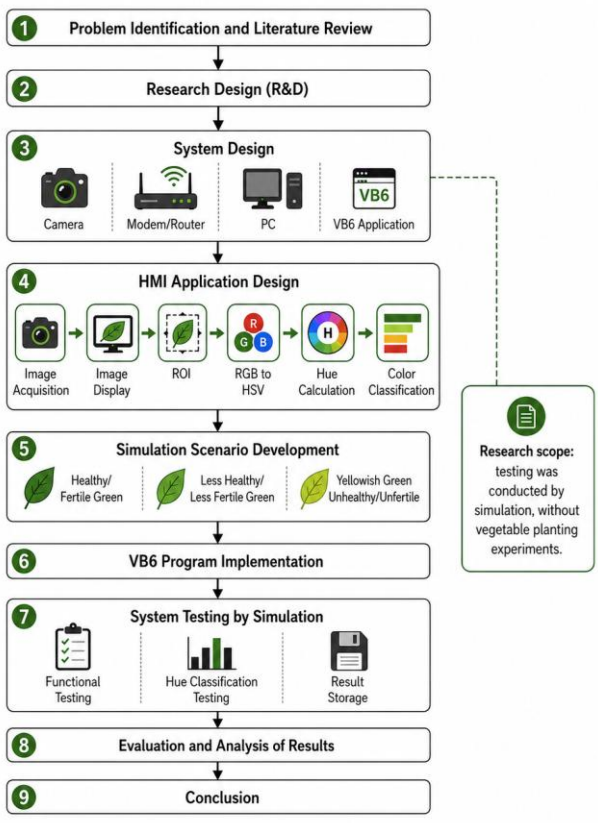


Fig. 1. Research design for the development of an HMI application for analyzing leaf greenness based on hue.

B. System Design

The system consists of a CCTV camera, a Wi-Fi modem/router, a PC, and an HMI application based on Visual Basic 6.0. The camera functions as an image acquisition device or visual input source. The modem/router serves as the communication medium between the camera and the PC. The PC runs the HMI application to display images, read RGB values, convert RGB to HSV, calculate hue, and display color categories in a simulated manner. The figure below shows the component setup. Fig. 2 shows a block diagram of the leaf green analysis system architecture, while Fig. 3 shows the system component setup. Table II lists the functions of each system component.

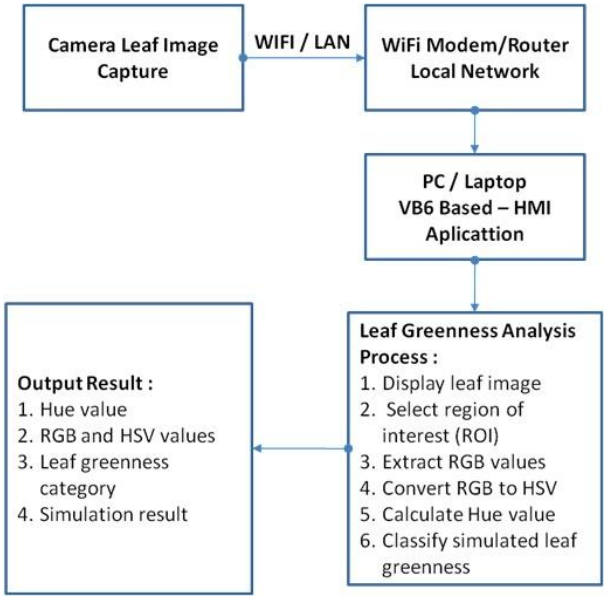


Fig. 2. System architecture block diagram.



Fig. 3. Concepts of configuring cameras, modems/routers, and data processing units.

TABLE II. FUNCTIONS OF SYSTEM COMPONENTS

Components	Functions
CCTV camera	Captures images or visual input for application testing.
Wi-Fi modem/router	Provides local communication between the camera and the PC.
PC/laptop	Runs the HMI application and performs color processing.
VB6 application	Loads images, extracts RGB values, converts to HSV, calculates hue, and displays simulated categories.

C. HMI Application Development

The HMI application was developed using Visual Basic 6.0. VB6 was chosen based on the availability of the programming environment and the ease of creating the user interface. Although VB6 is not a modern platform, this research remains relevant as an initial prototype and can be migrated in future research to Python, C#, or web-based applications. Table III shows the features included in the HMI, and Pseudocode 1 illustrates the pattern for building the HMI application program.

TABLE III. HMI APPLICATION FEATURES

HMI Module	Functions
Image acquisition/load	Capture or load an image/color sample.
Image display	Display the image in the PictureBox/HMI area.
ROI or area of analysis	Specify the color area to be analyzed.
RGB extraction	Read the R, G, and B values from the selected pixel.
HSV conversion	Convert RGB to Hue, Saturation, and Value.
Hue analysis	Calculate the average Hue as a color direction feature.
Classification	Assign a visual category based on Hue and Value rules.

Pseudocode 1. Methods for developing an HMI.

```
BEGIN PROGRAM

  INITIALIZE HMI APPLICATION
    Set image display area
    Set RGB output fields
    Set HSV output fields
    Set Hue output field
    Set leaf greenness category output
    Set data storage folder
    Set default status = "Ready"

  WHILE application is running DO

    IF user clicks "Load Image" OR "Capture Image" THEN
      Acquire image from camera or file
      Display image on HMI
      Set image status = "Image loaded"
    END IF

    IF user clicks "Select ROI" THEN
      User selects leaf area or color area
      Save selected ROI coordinates
      Display selected ROI on image
      Set ROI status = "ROI selected"
    END IF

    IF user clicks "Process" THEN

      IF image is not loaded THEN
     Display warning message: "Please load image first"
        STOP processing
      END IF

      IF ROI is not selected THEN
        Display warning message: "Please select ROI first"
        STOP processing
      END IF

      Read all pixels inside the selected ROI

      SET total_R = 0
```

```
SET total_G = 0
SET total_B = 0
SET total_Hue = 0
SET total_Saturation = 0
SET total_Value = 0
SET pixel_count = 0

FOR each pixel in ROI DO

   Read the RGB value of the pixel.
      R = Red value
      G = Green value
      B = Blue value

   Add R to total_R
   Add G to total_G
   Add B to total_B

   Normalize RGB values
      R' = R / 255
      G' = G / 255
      B' = B / 255

   Calculate Cmax
      Cmax = maximum value of R', G', B'

   Calculate Cmin
      Cmin = minimum value of R', G', B'

   Calculate Delta
      Delta = Cmax - Cmin

   Calculate Hue
      IF Delta = 0 THEN
         Hue = 0
      ELSE IF Cmax = R' THEN
         Hue = 60 × ((G' - B') / Delta)
         IF Hue < 0 THEN
            Hue = Hue + 360
         END IF
      ELSE IF Cmax = G' THEN
         Hue = 60 × (((B' - R') / Delta) + 2)
      ELSE IF Cmax = B' THEN
         Hue = 60 × (((R' - G') / Delta) + 4)
      END IF

   Calculate Saturation
      IF Cmax = 0 THEN
         Saturation = 0
      ELSE
         Saturation = Delta / Cmax
      END IF

   Calculate Value
      Value = Cmax

   Add Hue to total_Hue
   Add Saturation to total_Saturation
   Add Value to total_Value

   pixel_count = pixel_count + 1

END FOR

Calculate average RGB
   average_R = total_R / pixel_count
   average_G = total_G / pixel_count
   average_B = total_B / pixel_count

Calculate average HSV
   average_Hue = total_Hue / pixel_count
   average_Saturation = total_Saturation / pixel_count
   average_Value = total_Value / pixel_count

   Classify leaf greenness based on average_Hue

   IF average_Hue >= 45 AND average_Hue < 90 THEN
      category = "Yellowish green / unhealthy"
   ELSE IF average_Hue >= 90 AND average_Hue < 105 THEN
      category = "Pale green / less healthy"
   ELSE IF average_Hue >= 105 AND average_Hue <= 135 THEN
      category = "Healthy green"
   ELSE IF average_Hue > 135 AND average_Hue <= 160 THEN
      category = "Dark green / strong green"
   ELSE
      category = "Invalid or non-leaf color"
   END IF

   Display result on HMI
      Show average_R
      Show average_G
      Show average_B
      Show average_Hue
      Show average_Saturation
      Show average_Value
      Show leaf greenness category

   Set processing status = "Analysis completed"

END IF

IF user clicks "Save Result" THEN
   Save image
   Save ROI information
   Save average RGB values
   Save average HSV values
   Save Hue value
   Save leaf greenness category
   Display message: "Result saved successfully"
END IF

IF user clicks "Reset" THEN
   Clear image display
   Clear ROI selection
   Clear RGB output
   Clear HSV output
   Clear Hue output
   Clear category output
   Set status = "Ready"
END IF

IF user clicks "Exit" THEN
   Close application
END IF

END WHILE

END PROGRAM
```

D. RGB-to-HSV Conversion Algorithm

Each RGB pixel has R, G, and B values in the range of 0 to 255. Before being converted to HSV, these values are normalized to the range of 0 to 1 using Eq. (1). Afterward, the maximum, minimum, and chromatic difference values are calculated as shown in Eq. (2)- (4) [20].

$$R' = R/255,\ \ G' = G/255,\ \ B' = B/255 \tag{1}$$

$$Cmax = \max(R', G', B') \tag{2}$$

$$Cmin = \min(R', G', B') \tag{3}$$

$$\Delta = Cmax - Cmin \tag{4}$$

Hue is calculated according to the following rule:

- H = 0 if Δ = 0;
- H = 60 x (((G' - B') / Δ) mod 6), Cmax = R'
- H = 60 x (((B' - R') / Δ) + 2), Cmax = G'
- H = 60 x (((R' - G') / Δ) + 4), Cmax = B'

Moreover, Saturation is calculated as:

Saturation is defined as S = 0 when Cmax = 0, and S = Δ / Cmax otherwise. Value is defined as V = Cmax.

In this study, Hue is used as the primary feature for color direction. However, to avoid inconsistencies between light green and dark green, Value is used as a supporting attribute for brightness. Thus, color classification does not rely solely on Hue but also takes into account the darkness or brightness of the color. The categories "healthy," "less healthy," and "unhealthy" in this study are color-based simulation categories, not the result of physiological validation of the plants.

Furthermore, to ensure that value-based adjustments are explicit and reproducible, it is necessary to establish a formal classification rule that is applied consistently, as shown in Pseudocode 1 and in the verification example in the analysis section. Hue is assigned to each ROI as yellowish green ($45° \leq H < 90°$), pale green ($90° \leq H < 105°$), or green ($105° \leq H \leq 160°$). Within these green ranges, the average value is used to distinguish between normal green ($V > 0.55$), dark green ($0.40 \leq V \leq 0.55$), and very dark green ($V < 0.40$). Hue values in the range of 45–170° are considered invalid or not leaf colors, as shown in Table I.

E. Simulation Data and Evaluation Scenarios

The simulation data was compiled from two sources: 1) color samples representing visual leaf scenarios, namely greenish-yellow, pale green, normal green, dark green, and very dark green; 2) standard green colors from CSS named colors as additional test data with documented RGB values. This data was used solely as a computational test fixture, not as physiological plant data.

Evaluation was conducted through three types of testing: 1) black-box testing of the application's functionality; 2) algorithmic verification by comparing the application's output Hue, Saturation, and Value values against reference calculations; and 3) classification consistency evaluation, which assesses the alignment of the application's output with predefined simulated category rules. The metrics used were function success rate, absolute Hue error, and rule consistency accuracy. The absolute difference, in degrees, between the Hue result from the application and the reference value calculated using the RGB-to-HSV conversion formula.

IV. Result and Discussion

A. System and Application Implementation

The system implementation resulted in an HMI application prototype capable of loading images or color samples, displaying images on the interface, reading RGB values, converting RGB to HSV, calculating Hue, Saturation, and Value, and displaying color categories. This application is designed as a preliminary analysis tool that can be further developed using real leaf images once measurement and data collection facilities are available.

B. Results of Black-Box Functional Testing

Fig. 4 shows the display of the Leaf Green Analysis HMI, while Table IV shows that all tested functions operated as designed. The function success rate was 8 out of 8, or 100%. This value indicates the functional success of the software, not the physiological accuracy of the plant.

TABLE IV. Results of Black-Box Functional Testing

No	Test Scenario	Expected Results	Observation Results	Status
1	Open the application	The main HMI form appears	Displayed	Passed
2	Load/retrieve an image	The image appears on the HMI	Displayed	Passed
3	Select the analysis area	The selected color area is highlighted	Detected	Passed
4	RGB extraction	The R, G, and B values are displayed	Appears	Passed
5	HSV conversion	Hue, Saturation, and Value are displayed	Appears	Passed
6	Color classification	The simulation category is displayed	Displayed	Passed
7	Save results	The results are saved	Saved	Passed
8	Process without an image/ROI	The application displays a warning	Warning displayed	Passed

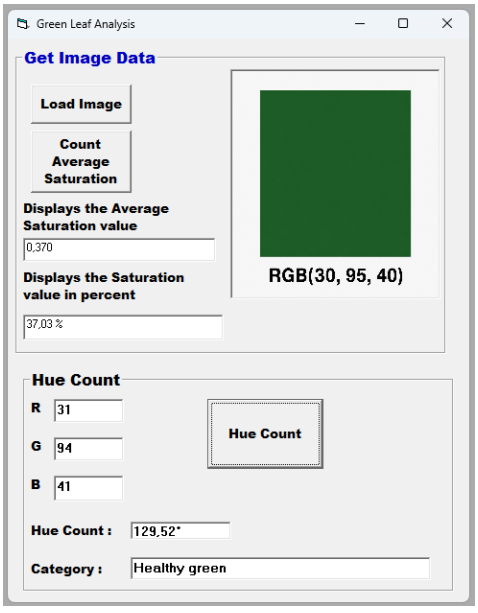


Fig. 4. HMI display: Leaf green analysis.

Fig. 4 shows the interface of the Green Leaf Analysis application, which was developed to analyze leaf green color based on RGB and HSV values. In this application, users can upload an image or a leaf color sample, and the program then reads the color values in RGB format. The example in the figure shows a color sample with RGB values of (30, 95, 40). Once the RGB values are obtained, the program calculates the Average

Saturation value, the Saturation value in percent, and the Hue value. The analysis results are then displayed as a leaf color category, namely “Healthy Green.”

Thus, this application functions as a simple HMI to assist in the simulation of leaf greenness analysis based on digital color features. To obtain the HSV values, follow these steps:

1) The initial values read from the image are: R = 30, G = 95, B = 40

a) Before being converted to HSV, the RGB values are normalized from the range 0–255 to 0–1:

- R! = (30 / 255) = 0.1176
- G! = (95 / 255) = 0.3725
- B! = (40 / 255) = 0.1569

b) Next, find the largest value, the smallest value, and the difference between them:

- Cmax = 0.3725
- Cmin = 0.1176
- Δ = Cmax − Cmin = 0.2549

c) Since the largest value is G’, the Hue formula used is:

- H = 60º x (((B! – R!) / Δ) + 2)
- H = 60º x (((0.1569 – 0.1176) / 0.2549) + 2)
- H = 60º x 2.154
- H = 129.23º

d) So the Hue value is approximately: H ≈ 129.52º

This value falls within the green range, so the app classifies it as "Healthy Green."

2) Calculating saturation:

- S = Δ / Cmax
- S = 0.2549 / 0.3725 = 0.684

If expressed as a percentage: S = 0.684 x 100 = 68.4%. This value indicates that the green color in the sample is quite strong—not too pale.

3) Calculating value, which indicates the brightness of a color:

- V = Cmax
- V = 0.3725
- V = 0.3725 x 100 = 37.25%

This means that the color is classified as very dark green because its brightness level is not very high.

C. Algorithmic Verification of RGB to HSV

Algorithmic verification was performed by calculating the HSV values of RGB samples using the reference formula and then comparing them with the application’s output. Table V shows the reference values used as the basis for testing. Since the color categories are simulated, the evaluation reported here focuses on the consistency of the algorithm and classification rules, not the accuracy of plant diagnosis.

As additional test data, several sRGB standard colors from CSS named colors were used to verify the consistency of the calculations with the documented RGB values. Table VI shows these color fixtures.

TABLE V. KEY SIMULATION SAMPLES AND THE RESULTS OF CONVERTING RGB TO HSV

Sample	Simulation category	R	G	B	Hue (degrees)	Saturation	Value	Color scale
S1	Greenish-yellow	145	160	55	68.57	0.656	0.627	Greenish-yellow
S2	Pale green	115	165	75	93.33	0.545	0.647	Pale green
S3	Normal green	70	150	55	110.53	0.633	0.588	Normal green
S4	Dark green	45	120	50	124.00	0.625	0.471	Dark green
S5	Very dark green	30	95	40	129.23	0.684	0.373	Very dark green

TABLE VI. SRGB/CSS COLOR FIXTURES FOR ADDITIONAL TESTING

Sample	Color names	RGB	Hue	Saturation	Value	Rule output
S6	ForestGreen CSS	RGB(34, 139, 34)	120.00	0.755	0.545	Dark green
S7	YellowGreen CSS	RGB(154, 205, 50)	79.74	0.756	0.804	Greenish yellow
S8	PaleGreen CSS	RGB(152, 251, 152)	120.00	0.394	0.984	Normal green
S9	DarkGreen CSS	RGB(0, 100, 0)	120.00	1.000	0.392	Very dark green
S10	OliveDrab CSS	RGB(107, 142, 35)	79.63	0.754	0.557	Greenish yellow
S11	LightGreen CSS	RGB(144, 238, 144)	120.00	0.395	0.933	Normal green
S12	Green CSS	RGB(0, 128, 0)	120.00	1.000	0.502	Dark green

D. Computational Comparison with RGB-Based Indices

In addition, simulation samples were also calculated using commonly used RGB-based indices, namely ExG, NGRDI, and the G/R ratio. This comparison is not intended as agronomic validation, but rather to demonstrate that the application can be developed to utilize not only Hue but also other color indices in future research. Moreover, the computational comparison of Hue with RGB-Based Indices on a Set of Key Simulation Samples is shown in Table VII.

TABLE VII. A COMPUTATIONAL COMPARISON OF HUE WITH RGB-BASED INDICES ON A SET OF KEY SIMULATION SAMPLES

Sample	R	G	B	ExG	NGRDI	G/R
S1	145	160	55	0.333	0.049	1.10
S2	115	165	75	0.394	0.179	1.43
S3	70	150	55	0.636	0.364	2.14
S4	45	120	50	0.674	0.455	2.67
S5	30	95	40	0.727	0.520	3.17

E. Discussion

The test results show that the VB6-based HMI application can perform basic leaf color analysis processes in a simulated environment. The application's main outputs include RGB values, hue, saturation, value, and color categories. Thus, this application can serve as an initial prototype to support digital image-based leaf greenness analysis. This research does not claim to provide physiological plant diagnoses; categories such as greenish-yellow, pale green, normal green, and dark green are visual categories based on color values. The research results are based on software evaluation and color simulation. Using Hue alone has limitations; some shades of normal green and dark green may have similar Hue values but differ in Value and Saturation. Therefore, Saturation and Value can be used as supporting attributes. Simulation results show that dark green is more accurately distinguished by a lower Value rather than solely by an increase in Hue. This addresses the inconsistencies in Hue classification found in previous versions of the manuscript.

Compared to the ExG index, NGRDI, and G/R ratio, Hue is easier to use for indicating color direction, while the RGB index better reflects the dominance of the green channel over the other channels. Since each feature carries different information, future research could combine Hue, Saturation, Value, ExG, NGRDI, and G/R to improve classification robustness once real-world leaf data becomes available. A limitation of this study is the use of simulated data; although simulated data is suitable for testing program logic, the results do not yet prove that the application can detect plant fertility or health in the field. Factors such as lighting, shadows, white balance, camera type, camera distance, and variations in leaf surface can alter RGB and HSV values. Therefore, this application should be positioned as an initial platform ready for validation in future research.

F. Implications of the Innovation and Limitations of Novelty

The innovation implications of this research are the availability of an HMI application design that can be used as an initial tool for education, demonstration, and testing of leaf color analysis logic. This application separates the contribution of the software from claims regarding plant physiology, thereby making the scope of the research clearer and more measurable. This approach is relevant for situations where researchers do not yet have a chlorophyll meter, SPAD meter, or real leaf dataset, but still wish to build the foundation for a computationally testable image analysis system.

A limitation of this study's novelty is that the classifications of "healthy," "less healthy," and "unhealthy" are still visual-simulatory in nature. Hue values cannot yet be used for actual plant health diagnosis without physiological validation. Additionally, the implementation in Visual Basic 6.0 is legacy; therefore, further research is recommended to migrate the algorithm to Python, C#, or web-based applications, as well as to incorporate validation using real leaf images, controlled lighting conditions, and additional features such as Saturation, Value, texture, or simple machine learning models.

V. CONCLUSION

This study produced a Visual Basic 6.0-based HMI application prototype to support the simulated analysis of leaf greenness using the Hue feature in the HSV color space. The application is capable of loading or capturing images, displaying images, reading RGB values, converting RGB to HSV, calculating Hue, Saturation, and Value, and assigning color categories based on simulated rules.

Black-box testing showed that all the application's main functions operated as designed with a 100% success rate in the tested scenarios. Algorithmic verification demonstrated that the classification output followed the established rules based on Hue and Value values. However, the results of this study only demonstrate the application's functional and computational feasibility, not the physiological validation of the plants.

Further research is needed to incorporate real leaf images, lighting controls, camera variations, and reference data such as SPAD, chlorophyll, leaf nitrogen, or expert assessments. Additionally, the application could be redeveloped using a more modern platform and combined with features such as Saturation, Value, texture, ExG, NGRDI, G/R ratio, or simple machine learning methods.

ACKNOWLEDGMENT

Thanks to the entire academic community of the Doctoral Program in Agro-Industrial Technology, Faculty of Agro-Industrial Technology, Padjadjaran University, Sumedang, West Java, Indonesia, as well as to the Research Center for Smart Mechatronics and Telecommunications, National Research and Innovation Agency of the Republic of Indonesia, Bandung, Indonesia, who have assisted in the publication of this article. We hope this article will continue to be cited and prove useful to related academic disciplines.

REFERENCES

[1] H. You, M. Zhou, J. Zhang, W. Peng, and C. Sun, "Sugarcane nitrogen nutrition estimation with digital images and machine learning methods," Sci. Rep., vol. 13, no. 1, pp. 1–12, 2023, doi: 10.1038/s41598-023-42190-2.

[2] X. Zhang, H. Yu, J. Yan, and X. Meng, "Study on the detection of chlorophyll content in tomato leaves based on RGB images,"

Horticulturae, vol. 11, no. 6, pp. 1–18, 2025, doi: 10.3390/horticulturae11060593.

[3] T. Luden, J. van Lieshout, S. L. Mehrem, B. L. Snoek, J. Willemse, and R. Offringa, "GreenLeafVI: A FIJI plugin for high-throughput analysis of leaf chlorophyll content," Physiol. Plant., vol. 177, no. 5, 2025, doi: 10.1111/ppl.70588.

[4] D. Li, C. Li, Y. Yao, M. Li, and L. Liu, "Modern imaging techniques in plant nutrition analysis: A review," Comput. Electron. Agric., vol. 174, p. 105459, May 2020, doi: 10.1016/j.compag.2020.105459.

[5] T. Haider et al., "A computer-vision-based approach for nitrogen content estimation in plant leaves," Agriculture, vol. 11, no. 8, pp. 1–19, 2021, doi: 10.3390/agriculture11080766.

[6] J. L. Croft, J. Arabian, J. M. Chen, and J. Shang, "Mapping within-field leaf chlorophyll content in agricultural crops for nitrogen management using Landsat-8 imagery," Precis. Agric., Springer, 2019, doi: 10.1007/s11119-019-09698-y.

[7] R. Adhikari and K. Nemali, "A novel method for estimating nitrogen stress in plants using smartphones," Horticulturae, vol. 6, no. 4, pp. 1–13, 2020, doi: 10.3390/horticulturae6040074.

[8] K. Berger et al., "Crop nitrogen monitoring: Recent progress and principal developments in the context of imaging spectroscopy missions," Remote Sens. Environ., vol. 242, p. 111758, 2020, doi: 10.1016/j.rse.2020.111758.

[9] R. O. Mendoza-Tafolla, R. E. Ontiveros-Capurata, P. Juarez-Lopez, I. Alia-Tejacal, V. Lopez-Martinez, and O. Ruiz-Alvarez, "Nitrogen and chlorophyll status in romaine lettuce using spectral indices from RGB digital images," Zemdirbyste, vol. 108, no. 1, pp. 79–86, 2021, doi: 10.13080/z-a.2021.108.011.

[10] X. Xiong, J. Zhang, D. Guo, L. Chang, and D. Huang, "Non-invasive sensing of nitrogen in plants using digital images and machine learning for Brassica campestris ssp. Chinensis L.," Sensors (Switzerland), vol. 19, no. 11, 2019, doi: 10.3390/s19112448.

[11] S. Qin et al., "Study on the nitrogen content estimation model of cotton leaves based on 'image-spectrum-fluorescence' data fusion," Front. Plant Sci., vol. 14, Mar. 2023, pp. 1–15, doi: 10.3389/fpls.2023.1117277.

[12] P. Bodor-Pesti, D. Taranyi, D. Á. Nyitrainé Sárdy, L. L. P. Nguyen, and L. Baranyai, "Correlation of the grapevine (Vitis vinifera L.) leaf chlorophyll concentration with RGB color indices," Horticulturae, vol. 9, no. 8, pp. 1–9, 2023, doi: 10.3390/horticulturae9080899.

[13] Mozilla Contributors, "CSS named-color: Standard colors in the sRGB color space," MDN Web Docs, Mozilla Developer Network. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Values/named-color

[14] H. Ge, G. Lv, Y. Qin, and M. Shen, "Assessing leaf nitrogen concentration in rice using RGB imaging: A comparative study at leaf, canopy, and plot scales," Front. Plant Sci., vol. 16, Aug. 2025, pp. 1–14, doi: 10.3389/fpls.2025.1599177.

[15] OpenCV Developers, "Changing color spaces," OpenCV Documentation. [Online]. Available: https://docs.opencv.org/5.0/

[16] Supriyono, "Software testing with the approach of black-box testing on the academic information system," Int. J. Inf. Syst. Technol., vol. 3, no. 2, pp. 227–233, 2020. [Online]. Available: https://ijist.universitypublication.com/article/view/blackbox-testing

[17] A. Agarwal, F. D. J. Colwell, V. A. C. Galvis, T. R. Hill, N. Boonham, and A. Prashar, "Assessing nutritional pigment content of green and red leafy vegetables by image analysis: Catching the 'red herring' of plant digital color processing via machine learning," Biol. Methods Protoc., vol. 10, no. 1, 2025, doi: 10.1093/biomethods/bpaf027.

[18] H. Zhang, Y. Ge, X. Xie, A. Atefi, N. K. Wijewardane, and S. Thapa, "High-throughput analysis of leaf chlorophyll content in sorghum using RGB, hyperspectral, and fluorescence imaging and sensor fusion," Plant Methods, vol. 18, no. 1, pp. 1–17, 2022, doi: 10.1186/s13007-022-00892-0.

[19] A. P. León, S. Z. Viña, D. Frezza, A. Chaves, and A. Chiesa, "Estimation of chlorophyll contents by correlations between SPAD-502 meter and chromameter in butterhead lettuce," Commun. Soil Sci. Plant Anal., vol. 38, no. 19–20, pp. 2877–2885, 2007, doi: 10.1080/00103620701663115.

[20] A. R. Smith, "Color gamut transformation pairs," Comput. Graph. (ACM), pp. 12–19, 1978. [Online]. Available: https://dl.acm.org/doi/10.1145/965139.807362

[21] Y. Ding, J. Chen, B. Xiong, J. A. Tuly, G. Cheng, and H. Ma, "An automated system for precision root and leaf cutting of green onions: intelligent design, modeling, and experimental validation," Measurement, Apr. 2026, doi: 10.1016/j.measurement.2026.120714.

[22] Y. Liu, L. Zheng, F. Kong, Y. Luo, H. Liang, S. Zhang, X. Tu, Z. Sheng, and G. Chen, "Active packaging film based on a green composite: fabrication and performance analysis of pineapple leaf fiber-reinforced PBAT," RSC Adv., vol. 16, no. 27, pp. 24296–24305, May 2026, doi: 10.1039/d5ra10078j.

[23] T. Y. Kim, H. J. Woo, and T. M. Yoon, "Enhanced protective effect of combined treatment with green tea seed and leaf extract against HCl and ethanol-induced gastritis and gastric ulcer in rats," Heliyon, vol. 12, no. 9, p. e44938, May 2026, doi: 10.1016/j.heliyon.2026.e44938.

[24] A. D. Nusseif, R. S. Sabry, N. A. Hussain, and A. N. Majeed, "Novel green TiO_2 nanoparticles from Corchorus olitorius leaf extract for solar and UV photocatalytic degradation of methylene blue and tetracycline," Next Mater., vol. 13, p. 103202, Oct. 2026, doi: 10.1016/j.nxmate.2026.103202.

[25] T. Manimegalai, S. I. K. Selvam, L. A. Devi, and D. K. S. Kumari, "Green zinc oxide nanoparticles from Tagetes erecta (L.) leaves reduce Meloidogyne incognita-induced stress in Solanum lycopersicum (L.)," Physiol. Mol. Plant Pathol., vol. 145, p. 103382, Sep. 2026, doi: 10.1016/j.pmpp.2026.103382.

[26] C. C. M. Figueiredo, F. O. Granero, J. L. B. Junior, L. P. Silva, M. F. Minicucci, and R. M. G. da Silva, "Spinach leaf biomass valorization through coenzyme Q10 recovery and its application in silver nanoparticles green synthesis: Physicochemical characterization and antioxidant activity," Food Bioprod. Process, vol. 159, pp. 712–716, Sep. 2026, doi: 10.1016/j.fbp.2026.07.028.

[27] H. Krishnan, B. Lakshmipathy, J. Ramaswamy, I. Nallamuthu, M. Dubey, B. Xu, C. K. Reddy, and M. Kumar, "Green extraction of RuBisCO-rich leaf protein for sustainable food systems: process efficiency, functional quality, and sustainability assessment," Food Chem., Aug. 2026, doi: 10.1016/j.foodchem.2026.149914.

[28] S. T. Debelle, Z. Barati, S. Romuli, D. Y. Mengesha, A. Tolera, and J. Müller, "Protein-enriched 'green' gari from optimized cassava root-leaf mixtures using Box-Behnken design," Appl. Food Res., vol. 6, no. 1, p. 102244, Jun. 2026, doi: 10.1016/j.afres.2026.102244.

[29] H. S. Kusuma, K. D. Az-Zahra, and A. E. Arista, "Investigation and optimization of patchouli leaf waste extract (PLWE) as a green corrosion inhibitor for aluminum in HCl via Box–Behnken design," Hybrid Adv., vol. 13, p. 100661, Jun. 2026, doi: 10.1016/j.hybadv.2026.100661.

[30] D. D. Upadhyay, S. Singh, V. Mishra, P. Yadav, S. K. Shrivastava, and G. Pandey, "Green fabrication of ZnO@ZnS binary nanocomposite using Butea monosperma leaf extract: antibacterial activity and antioxidant enzyme modulation in Spinacia oleracea," Plant Nano Biol., vol. 17, p. 100322, Aug. 2026, doi: 10.1016/j.plana.2026.100322.

An Architecture-Aware Flutter Code Generation Framework for Enforcing Clean Architecture and Behavioral Design Patterns

Desy Intan Permatasari[1]*, Adam Shidqul Aziz[2], Maulidan Bagus Afridian Rasyid[3],
Ahmad Jarir At-Thobari[4], Umi Sa'adah[5], Nailussa'ada[6], Muhammad Fajrul Falah Subakti[7]
Informatics and Computer Engineering Department, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia[1, 2, 5, 6, 7]
SimHive Group, Surabaya, Indonesia[3]
AgileTeknik Network, Surabaya, Indonesia[4]

Abstract*—Large-scale Flutter applications can accumulate direct inter-module dependencies, inconsistent responsibility boundaries, and inflexible data-access implementations as their functionality grows. Conventional scaffolding approaches primarily automate project structures and repetitive code, while architectural relationships commonly remain dependent on developer-level implementation decisions. This study proposes an architecture-aware Flutter code generation framework that transforms JavaScript Object Notation (JSON)-based application specifications into modular source code while systematically applying Clean Architecture, Mediator, and Strategy rules. The main contribution lies not in combining these established architectural concepts, but in encoding their architectural roles as repeatable specification-to-code transformation rules. The framework was evaluated using three application specifications containing one, two, and eight functional modules. All three applications were successfully generated, executed, and built into installable Android APKs. In the evaluated Login and Register interaction, two direct cross-module relationships observed in the legacy architecture were replaced by mediator-based communication, resulting in no direct Login and Register reference in the generated structure. The generated architecture also represented REST API and SQLite data providers through a common Strategy-based abstraction. These findings demonstrate operational feasibility and provide structural evidence that the intended architectural rules were incorporated into the evaluated generated artifacts. However, the evaluation is limited to the supported JSON schema, predefined generation templates, three application specifications, and the evaluated architectural relationships; therefore, the results should not be interpreted as general evidence of long-term maintainability, scalability, or superiority over other Flutter code generators.

Keywords*—Architecture-aware code generation; specification-driven development; flutter; clean architecture; mediator pattern; strategy pattern; software modularity

I. Introduction

Flutter has emerged as a widely adopted framework for cross-platform mobile application development because it enables applications for multiple platforms to be developed from a shared codebase while supporting efficient development processes and near-native performance [1]. However, as mobile applications grow in functional scope and architectural complexity, maintaining a consistent source-code structure becomes increasingly challenging. Large-scale applications may accumulate direct dependencies among functional modules, unclear responsibility boundaries, and dependency-related problems that increase the difficulty of software maintenance and evolution [2], [3].

Clean Architecture addresses these challenges by separating software responsibilities into distinct layers and directing dependencies toward stable abstractions [4]. Previous studies have shown that its application can improve responsibility separation, source-code organization, and maintainability in mobile software projects [5], [6]. Nevertheless, Clean Architecture primarily defines responsibility boundaries and dependency directions. It does not by itself prescribe a uniform mechanism for coordinating communication among functional modules or for representing interchangeable operational behaviors, such as alternative data-access mechanisms.

Behavioral design patterns provide complementary mechanisms for addressing these concerns. The Mediator pattern centralizes communication among participating components and reduces the need for direct references between them [7]. Its implementation in software systems has demonstrated how component interactions can be coordinated through an intermediary rather than through direct object-to-object dependencies [8]. The Strategy pattern encapsulates alternative algorithms or behaviors behind a common interface, allowing a concrete implementation to be selected or replaced without modifying its client [9]. In a modular Flutter application, these patterns can support controlled inter-module communication and flexible selection of data providers, including REST application programming interface (API) and SQLite implementations.

Existing Flutter studies have generally applied Clean Architecture manually within application-specific projects. For example, Clean Architecture has been implemented in a Flutter-based tourism application by manually defining the presentation, domain, and data structures required by the application [10].

Related software-engineering research has also emphasized that reducing dependencies among software components is

*Corresponding author.

important for architectural scalability and long-term system evolution [11], while separation of concerns contributes to the management of software complexity [12]. A reusable Flutter Clean Architecture package has additionally been proposed to provide predefined architectural components and assist developers in separating business logic from platform-specific implementation [13].

Although these approaches support architectural adoption, they still require developers to define functional modules, configure their relationships, implement inter-module communication, select data-access mechanisms, and maintain architectural consistency at the source-code level. Reusable packages and manually implemented architectures can reduce initial development effort, but they do not necessarily ensure that the same architectural constraints are applied consistently across all modules and application versions. Similarly, conventional scaffolding tools may generate folders, classes, and repetitive source-code artifacts without systematically enforcing the relationships among the generated components.

Within the literature reviewed in this study, limited attention has been given to transforming declarative Flutter application specifications into multi-module applications while systematically enforcing responsibility separation, mediator-based inter-module communication, and strategy-based data-provider variation during source-code generation. The research gap addressed in this study is therefore not the absence of Clean Architecture, Mediator, or Strategy as individual software-engineering concepts, since these concepts are already well established. Rather, the unresolved issue concerns their systematic enforcement as part of an automated specification-to-code generation process. Without such enforcement, architectural conformance remains dependent on developer expertise and local implementation decisions, potentially resulting in inconsistent dependency structures and architectural drift.

To address this gap, this study proposes an architecture-aware Flutter code generation framework that receives application specifications in JavaScript Object Notation (JSON) format and transforms them into modular Flutter source code. The JSON input serves as a blueprint that defines application-level properties, pages, routes, functional modules, and selected data providers. During generation, the framework applies Clean Architecture rules to organize software responsibilities, Mediator rules to coordinate communication among functional modules, and Strategy rules to represent REST API and SQLite data providers through interchangeable abstractions.

The principal novelty of the proposed framework does not lie merely in combining Clean Architecture with the Mediator and Strategy patterns. Its scientific contribution lies in encoding these architectural decisions as repeatable specification-to-code generation rules. Consequently, responsibility boundaries, communication mechanisms, and data-provider abstractions are established during source-code generation rather than being left entirely to post-generation manual implementation. This mechanism is intended to produce consistent architectural structures across modules with different functional responsibilities.

This study makes three main contributions. First, it defines a JSON-based specification-to-architecture mapping that transforms application properties, pages, routes, functional modules, and data-provider configurations into organized Flutter source-code artifacts. Second, it implements an automated generation mechanism that systematically applies Clean Architecture boundaries, mediator-based inter-module communication, and strategy-based data-provider variation during source-code generation. Third, it provides an empirical feasibility and structural evaluation using three application specifications with increasing modular complexity, including operational generation and build observations, a legacy–generated architectural comparison, and inspection of the generated Mediator and Strategy structures.

The evaluation examines whether the framework can successfully generate, execute, and build Flutter applications from the tested JSON specifications. Structural architectural characteristics are assessed by comparing the available legacy and generated application structures, examining direct functional-module relationships, and inspecting whether the generated source structure incorporates the intended Mediator and Strategy mechanisms. The results are interpreted as evidence of operational feasibility and architectural conformance within the evaluated scenarios. They are not considered complete proof of long-term maintainability, scalability, or testability because these broader quality attributes require larger application samples, repeated experiments, additional software-quality measurements, and longitudinal maintenance evaluation.

The remainder of this study is organized as follows. Section II critically reviews previous studies on Clean Architecture, behavioral design patterns, Flutter architectural approaches, code generation, and software-quality evaluation. Section III describes the proposed framework, JSON specification, architectural generation rules, experimental scenarios, and evaluation procedure. Section IV presents and discusses the operational and structural evaluation results. Section V discusses the threats to validity and limitations of the study. Finally, Section VI presents the conclusions and directions for future research.

II. Related Work

Clean Architecture has been applied in mobile software development to improve separation of concerns and control dependency directions. Rachovski et al. [4] used Clean Architecture principles to organize the design and implementation of a mobile platform, while Santiago-Salazar and Rico-Bautista [5] examined its impact on the performance and maintainability of native Android applications. In the Flutter context, Sinatria et al. [6] implemented Clean Architecture by separating presentation, domain, and data responsibilities. These studies indicate that layered architectural separation can clarify responsibility boundaries and improve source-code organization. However, the architectural structures in these studies were implemented manually within individual application projects, meaning that architectural consistency remained dependent on developer knowledge and implementation decisions.

Behavioral design patterns provide complementary mechanisms for managing software interactions and behavioral variation. The Mediator pattern centralizes communication among participating components, thereby reducing the need for direct references between them [7]. Petrova and Varbanov [8] demonstrated mediator-based communication and command handling using the MediatR library in a .NET environment. Meanwhile, the Strategy pattern encapsulates alternative algorithms or behaviors behind a common interface, allowing an implementation to be selected or replaced without modifying its client [9]. Although these studies establish the architectural roles of Mediator and Strategy, the patterns are generally applied manually at the source-code level. Consequently, their implementation may vary among modules, developers, and application versions.

Wijayanto and Sejati [10] applied Clean Architecture to a Flutter-based mobile tourism application and reported that the separation of presentation, domain, and data responsibilities supported application modularity and development. Nevertheless, the architectural components and their relationships were manually defined for the specific application rather than generated from a declarative specification. In a broader software-architecture context, Aljaloud and Razzaq [11] demonstrated the importance of reducing dependencies when restructuring monolithic software systems, while Idate et al. [12] emphasized separation of concerns as a mechanism for managing software complexity. These studies reinforce the importance of modularity and dependency management but do not provide an automated mechanism for generating modular Flutter applications.

A more directly relevant reusable solution was introduced by Boukhary and Colmenares [13] through the Flutter Clean Architecture Package. The package provides reusable architectural components that help developers separate business logic from platform-specific implementation. This approach reduces the effort required to establish a Clean Architecture structure in Flutter projects. However, a reusable architectural package differs from an architecture-aware code generator. Developers must still define the functional modules, configure dependencies, implement communication among modules, and determine how alternative data providers are integrated. Therefore, the package supports architectural implementation but does not automatically transform an application specification into a complete multi-module Flutter project with generated Mediator and Strategy structures.

Stirgwolt et al. [14] demonstrated that modular architectures can support system evolution by controlling relationships among components. Sangwan et al. [15] further emphasized that architectural dependency analysis is necessary because uncontrolled dependencies may increase change propagation and complicate software evolution. Sun et al. [16] investigated the relationship between dependency injection and software maintainability, demonstrating the relevance of dependency management to software-quality assessment.

Recent model-driven software-engineering studies provide a broader perspective on systematic and architecture-aware software generation. Lano et al. [17] proposed an agile model-driven re-engineering approach that uses software abstraction, semantic models, transformation, refactoring, and rearchitecting to support legacy-system modernization and platform migration. Their evaluation included practical re-engineering cases involving legacy systems and comparison of program-translation effectiveness, demonstrating how explicit model-driven transformations can support automated restructuring and target-code production.

Jiménez-Navajas et al. [18] proposed a model-to-text code-generation technique in which extended Unified Modeling Language (UML) design models are transformed into executable source code. Their approach uses explicit Model-to-Text transformations to generate Python and Qiskit code for classical-quantum systems and was evaluated through a multi-case study involving seven software systems. Although the target domain differs from Flutter mobile applications, the study provides a relevant example of specification-driven generation in which transformation rules and generated artifacts are explicitly defined and empirically evaluated.

Recent studies published in applied computer-science venues provide additional perspectives relevant to the present work. Tiali et al. [19] employed a model-driven engineering approach in which explicit transformation rules systematically map business-process representations into generated executable smart-contract artifacts. Although their source and target technologies differ from the present study, their work illustrates the importance of making transformation rules explicit when automated generation is claimed. In the mobile-software context, Yataco et al. [20] demonstrated the practical use of Clean Architecture in an implemented mobile system, showing the continuing relevance of responsibility separation and modular architectural organization in application development. From a software-quality perspective, Abufakher [21] empirically investigated GoF design patterns using multiple object-oriented quality metrics, illustrating the importance of explicitly defined measurement procedures when quantitative structural metrics are reported.

These studies strengthen the methodological positioning of the present work because they demonstrate that model- or specification-driven software generation can encode explicit transformation rules rather than relying exclusively on manual implementation. However, none of these studies addresses the generation of modular Flutter applications in which Clean Architecture organization, mediator-based inter-module communication, and strategy-based data-provider variation are jointly incorporated into the generated source structure.

Collectively, these studies demonstrate that architectural quality can be examined from multiple structural perspectives, including cohesion, coupling, modular organization, and dependency relationships. They also indicate that individual static metrics capture only selected characteristics of a software architecture and should not be interpreted as complete measures of long-term maintainability, scalability, testability, or developer effort. Accordingly, the present study focuses its empirical structural evaluation on directly observable architectural relationships in the evaluated artifacts, particularly direct functional-module dependencies and the generated Mediator and Strategy structures.

Table I compares representative related approaches according to their target platforms, input mechanisms, architectural support, levels of automation, evaluation methods, and remaining limitations. The comparison distinguishes manually implemented Clean Architecture applications, reusable architecture packages, model-driven generation approaches, and the architecture-aware Flutter code generation framework proposed in this study.

As shown in Table I, existing Flutter studies mainly apply Clean Architecture manually or provide reusable architectural components whose final configuration remains developer-controlled. In contrast, recent model-driven approaches [17], [18] demonstrate that explicit software representations can be transformed systematically into generated artifacts. The proposed framework combines these two directions by using a declarative JSON specification to generate Flutter source code while incorporating predefined Clean Architecture organization, Mediator-based inter-module communication, and Strategy-based data-provider variation.

Nevertheless, Table I should be interpreted as a capability-level comparison rather than a direct performance benchmark. The reviewed studies differ in platform, input representation, transformation objectives, generated technologies, and evaluation procedures. Therefore, the empirical evaluation in this study uses the available legacy application as an internal structural baseline, while a controlled comparison with functionally equivalent external Flutter generators remains future work.

TABLE I. COMPARISON OF RELATED ARCHITECTURAL APPROACHES

Study	Platform / Input	Architectural Support	Automation Level	Evaluation	Remaining Limitation
Wijayanto and Sejati [10]	Flutter; manually implemented requirements	Clean Architecture for modular mobile	Manual application-specific implementation	Functional application case study	Not generated from declarative specification
Boukhary and Colmenares [13]	Flutter; reusable package and developer configuration	Reusable Clean Architecture components	Package-assisted; assembly remains developer-controlled	Package demonstration/application implementation	No complete specification-driven multi-module generation
Lano et al. [17]	Legacy software and semantic models	Refactoring, restructuring, rearchitecting, migration	Automated/model-driven transformation	Real-world legacy cases and translation evaluation	Different target problem/platform; no Flutter Mediator/Strategy generation
Jiménez-Navajas et al. [18]	Extended UML models	Model-driven source-code generation	Automated Model-to-Text transformation	Seven-system multi-case study	Different target domain/output technology; no Flutter architecture-pattern enforcement
Proposed framework	Flutter; JSON application specifications	Clean Architecture organization, Mediator communication, Strategy data-provider variation	Automated specification-to-code generation	One-, two-, and eight-module scenarios; execution/build validation; structural evaluation	Limited to supported JSON schema, templates, modules, and evaluated scenarios

III. METHODOLOGY

A. Framework Overview and Architecture

This study evaluates a Clean Code Generator developed as part of the CleanApp environment. The generator receives an application specification in JSON format and produces Flutter source-code artifacts using predefined generation templates. The JSON configuration describes the structure and functional elements of the target application, while the generation mechanism incorporates architectural structures based on Clean Architecture, Mediator, and Strategy. The resulting Flutter project is subsequently executed, built into an Android application package (APK), and inspected through operational and structural architectural evaluation.

B. JSON-Based Application Specification

The Clean Code Generator receives its input as a JSON configuration produced by the CleanApp Builder. The configuration describes the application structure to be generated, including application-level information, page definitions, navigation routes, data-provider configurations, optional features, and application branding information. The JSON specification therefore serves as a declarative representation of the target application, while the generator translates the supplied configuration into the corresponding Flutter source-code structure.

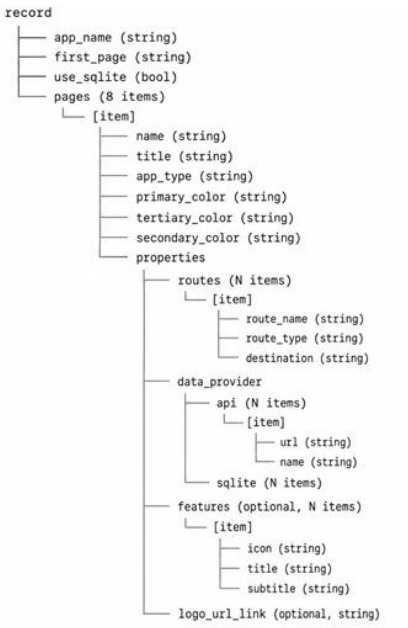


Fig. 1. Structure of the JSON application.

As shown in Fig. 1, the specification contains three application-level elements: app_name, first_page, and use_sqlite. The pages collection contains the definitions of the

functional pages included in the generated application. Each page contains descriptive attributes, including its name, title, application type, and interface colors. The properties element contains the configuration required to determine navigation, data access, optional features, and application branding.

Navigation is represented through the routes collection, in which each route contains route_name, route_type, and destination. Data-access requirements are represented by the data_provider element, which can contain API- and SQLite-based configurations. An API entry contains a url and name, while the SQLite configuration can contain multiple entries depending on the application specification.

The high-complexity configuration can additionally contain optional features, consisting of icon, title, and subtitle, as well as an optional logo_url_link.

TABLE II. JSON SPECIFICATION CHARACTERISTICS OF THE EXPERIMENTAL SCENARIOS

Configuration element	Low Complexity	Medium Complexity	High Complexity
pages	1 item	2 items	8 items
routes	2 items	N items	N items
data_provider.api	0 items	0 items	N items
data_provider.sqlite	0 items	0 items	N items
features	Not shown	Not shown	Optional, N items
logo_url_link	Present	Not shown in displayed schema	Optional

Note: N denotes a variable number of items in the displayed JSON schema rather than a fixed experimental count.

The number of elements varies according to the application scenario. Therefore, the JSON specification does not directly encode a fixed application implementation; instead, it defines the application elements that must be interpreted by the generator. The generator subsequently applies its predefined architectural rules and source-code templates to produce the corresponding Flutter project.

Table II highlights the differences among the JSON structures used in the three experimental scenarios. The low-complexity configuration contains a single page and two navigation routes, whereas the medium-complexity configuration contains two pages and supports a variable number of routes. In both displayed configurations, the API and SQLite provider collections contain no items. The high-complexity specification extends this structure to eight pages and supports multiple routes, API and SQLite entries, optional feature definitions, and an optional logo URL. Thus, experimental scenarios increase not only in the number of generated functional components but also in the richness of the configuration processed by the generator.

C. Architecture-Aware Code Generation Mechanism

The proposed framework transforms the JSON specification into Flutter source-code artifacts using predefined code templates and architectural rules. Clean Architecture provides the structural organization of the generated application, while the Mediator and Strategy patterns define specific relationships among generated components. The Mediator mechanism is used to coordinate communication among functional modules without requiring direct module-to-module references, whereas the Strategy mechanism is used to organize alternative data-access implementations behind a common abstraction. Thus, the generator does not merely produce project folders and repetitive source code; it also embeds predefined architectural relationships into the generated project.

In the generated architecture, each functional module contains a controller that acts as the principal coordination point for the corresponding module. Interactions that involve other functional modules are routed through the generated mediator structure, while data-access alternatives such as REST API and SQLite are represented through Strategy-based components. This mechanism allows the same architectural conventions to be repeatedly applied to different application specifications and functional modules.

The architecture-aware generation mechanism can be summarized as a set of specification-to-code transformation rules. These rules determine how application elements and architectural conditions are translated into generated source-code structures. Table III summarizes the principal rules implemented by the framework and the architectural structures produced by each rule.

TABLE III. SPECIFICATION-TO-CODE ARCHITECTURAL GENERATION RULES

Specification or Architectural Condition	Generation Rule	Generated Structure / Architectural Effect
Functional page or module definition	Generate the corresponding functional-module source structure, including its controller, using the predefined architectural templates	Module responsibilities are organized according to the predefined Clean Architecture structure
Interaction involving different functional modules	Route cross-module communication through the generated Mediator component rather than generating direct module-to-module references	Mediator-based inter-module coordination
REST API data-provider configuration	Generate the API-specific data-provider implementation behind the common data-provider abstraction	Strategy-based representation of REST API access
SQLite data-provider configuration	Generate the SQLite-specific data-provider implementation behind the same data-provider abstraction	Strategy-based representation of SQLite access

In this study, architecture-aware generation refers to a code-generation process in which predefined architectural rules are applied during the transformation from the JSON specification to source-code artifacts, rather than generating only syntactic scaffolding. Architectural enforcement refers to the generation-time instantiation of these selected rules in the produced source structure, so that the corresponding architectural decisions are not left entirely to post-generation developer implementation. Architectural conformance is operationally defined as the observable agreement between the generated artifacts and the specific architectural rules evaluated in this study, namely the predefined module organization, Mediator-based routing of the evaluated cross-module interaction, and Strategy-based representation of REST API

and SQLite data providers. This definition is limited to the inspected architectural relationships and does not constitute formal verification of all Clean Architecture dependency constraints.

These rules distinguish the proposed framework from conventional scaffolding that primarily generates repetitive project artifacts. In the proposed approach, the generator also determines selected architectural relationships during source-code generation. Consequently, module organization, cross-module communication, and data-provider variation are instantiated according to predefined architectural rules rather than being left entirely to post-generation manual implementation.

D. Code Generation Workflow

Fig. 2 summarizes the specification-to-code generation and validation workflow. The code-generation process begins when the Clean Code Generator receives a JSON configuration produced by the CleanApp Builder.

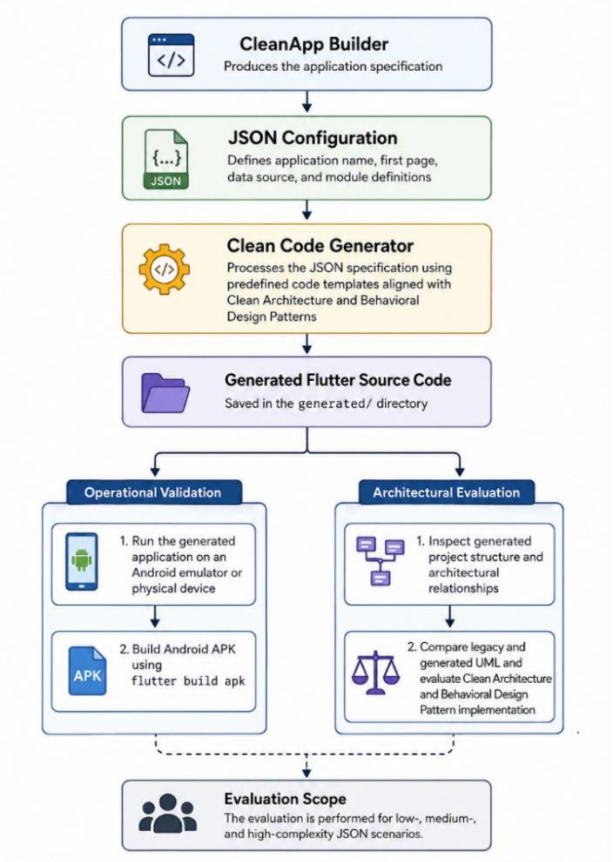


Fig. 2. Workflow of the specification-to-code generation and validation process.

The framework first reads the application-level configuration and identifies the pages, navigation routes, functional modules, and data-provider requirements defined in the specification. The corresponding source-code templates are then selected and used to generate the required Flutter project artifacts. During this process, the predefined architectural structures for Clean Architecture, Mediator-based communication, and Strategy-based data access are incorporated into the generated source code.

After generation, the resulting files are organized into a standalone Flutter project. The generated project is subsequently executed in the development environment or on an Android device to verify that the application can run successfully. Finally, the project is built into an Android APK and its source-code structure is inspected for architectural evaluation. This workflow therefore covers the complete process from declarative application specification to generated, executable, and structurally evaluated Flutter application.

E. Experimental Design and Environment

The framework was evaluated using three JSON-based application specifications representing increasing levels of functional complexity. The low-complexity scenario contained one functional module implementing a registration form. The medium-complexity scenario contained two functional modules, Setting and User. The high-complexity scenario incorporated all functional modules available in the evaluated framework configuration. These three scenarios were selected to examine whether the same generation mechanism could process application specifications with different numbers of functional components while preserving the intended project structure.

Table IV summarizes the evaluated scenarios, including their relative configuration scale, number of functional modules, and functional scope.

TABLE IV. EXPERIMENTAL SCENARIOS

Scenario	Scenario Scale	Modules	Functional Scope
S1	Low	1	Registration
S2	Medium	2	Setting and User
S3	High	8	Login, E-commerce List, E-commerce Square, Note, Onboarding, Register, Settings, and User

The experiments were conducted from August 2024 to June 2025. The primary development machine was equipped with an 11th-generation Intel Core i5 processor, 16 GB of RAM, and a 512 GB SSD. The software environment consisted of Flutter Software Development Kit (SDK) 3.24.5, Dart SDK 3.5.4, Android Studio Jellyfish 2023.3.1, and Visual Studio Code. The generated applications were additionally validated using two Android devices: one device equipped with an Exynos 9611 chipset, 4 GB of RAM, and Android 10, and another equipped with a MediaTek Dimensity 700 chipset, 8 GB of RAM, and Android 13.

In this experiment, the terms low, medium, and high complexity are used as descriptive scenario labels reflecting the increasing number of functional modules and the relative richness of the corresponding JSON specifications. They should not be interpreted as formal measurements of software complexity, dependency density, interaction complexity, generated-code volume, architectural depth, or cyclomatic complexity. The three scenarios were selected to examine whether the same generation mechanism could process

specifications with different numbers of functional components while preserving the intended project structure.

F. Evaluation Procedure

The evaluation consisted of two complementary stages: operational validation and structural architectural evaluation. Operational validation examined whether each JSON specification could be successfully processed into Flutter source code, whether the generated project contained the expected directory and source-code structure, whether the application could be executed on the experimental environment, and whether an Android APK could be successfully built. Generation time, APK build time, and APK size were recorded as descriptive operational characteristics.

Structural architectural evaluation examined whether the generated source code exhibited the intended architectural relationships. The generated application structure was represented using UML class diagrams and compared with the available legacy application architecture. The comparison focused on direct relationships between functional modules and on whether the generated application routed cross-module communication through the Mediator component. The generated Strategy structure was also inspected to verify the representation of REST API and SQLite data-provider implementations through a common abstraction.

For the structural comparison, a direct functional-module dependency was counted when a class belonging to one functional module directly referenced a class belonging to another functional module without an intermediary architectural abstraction. Dependencies routed through the generated Mediator abstraction were treated as mediated relationships and were therefore not counted as direct functional-module dependencies. The comparison was restricted to the evaluated Login–Register interaction for which corresponding structural evidence was available in both the legacy and generated architectures.

Accordingly, the structural evaluation focuses on architectural relationships that can be directly inspected from the available generated and legacy artifacts. The evaluation does not attempt to provide a comprehensive quantitative assessment of software maintainability or overall architectural quality.

IV. Results and Discussion

A. Framework Validation

The operational validation was conducted using three application-generation scenarios with different levels of functional complexity. The scenarios consisted of low-complexity applications with a single module, medium-complexity applications with two modules, and high-complexity applications containing all available functional modules. For each scenario, the framework was required to successfully generate source code and produce a buildable Flutter application.

Operational success in this evaluation indicates that the generated projects could be produced, executed, and built successfully under the tested configurations. However, successful execution and APK construction do not by themselves establish functional correctness with respect to every behavioral requirement represented in the JSON specifications. A systematic requirements-to-behavior verification was outside the scope of the present evaluation.

TABLE V. Framework Validation

Complexity	Modules	Code Generation	Application Execution	APK Build
Low	1	Successful	Successful	Successful
Medium	2	Successful	Successful	Successful
High	8	Successful	Successful	Successful

Table V summarizes the operational validation results. All three evaluated specifications were successfully generated, executed, and built into Android APKs. Because each scenario was evaluated once, these observations demonstrate operational feasibility within the tested configurations rather than a generalized reliability rate.

B. Structural Comparison of Legacy and Generated Architectures

The legacy and generated application structures were compared to examine differences in direct functional-module relationships after the introduction of mediator-based coordination. The comparison focuses on the evaluated Login–Register interaction, for which corresponding structural evidence is available in both architectures.

TABLE VI. Structural Comparison of the Evaluated Legacy and Generated Interaction

Structural Aspect	Legacy Architecture	Generated Architecture
Evaluated interaction	Login ↔ Register	Login ↔ Register through Mediator
Direct functional-module dependencies	2 directed relationships	0 direct relationships
Inter-module coordination	Direct cross-module references	Mediator-based coordination
Architectural interpretation	Modules directly reference one another	Direct references replaced by mediated communication

Table VI summarizes the structural comparison of the evaluated Login–Register interaction. In the legacy architecture, two directed cross-module relationships were observed between the corresponding components. In the generated architecture, no direct Login–Register relationship was observed because the interaction is routed through the Mediator component. Thus, the direct functional-module dependency count decreases from two to zero for this evaluated interaction. This result provides structural evidence of reduced direct coupling, although the generated modules remain dependent on the Mediator abstraction.

TABLE VII. Operational Measurements

Complexity	Modules	Generation Time (ms)	Build Time (s)	APK Size (MB)
Low	1	1,311	38.3	49.0
Medium	2	1,351	302.5	49.8
High	8	1,474	249.7	65.8

Table VII presents the descriptive operational measurements of the three evaluated specifications. Generation time increased from 1,311 ms for the single-module specification to 1,474 ms for the eight-module specification. APK size increased from 49.0 MB to 65.8 MB across the same configurations. In contrast, build time did not exhibit a monotonic relationship with the number of modules, with recorded values of 38.3 s, 302.5 s, and 249.7 s. Because each specification was evaluated once, these measurements are reported descriptively and are not interpreted as statistically validated evidence of runtime or build scalability.

The non-monotonic build-time pattern cannot be attributed to configuration scale from the present measurements alone. Potential contributing factors may include build-system state, dependency resolution, incremental-build or caching conditions, and background execution variability. Because these factors were not independently controlled or recorded during the experiment, the observed differences are treated as descriptive tooling and execution variability rather than evidence of a systematic relationship between module count and build time. Repeated measurements under controlled build conditions are required to distinguish these effects.

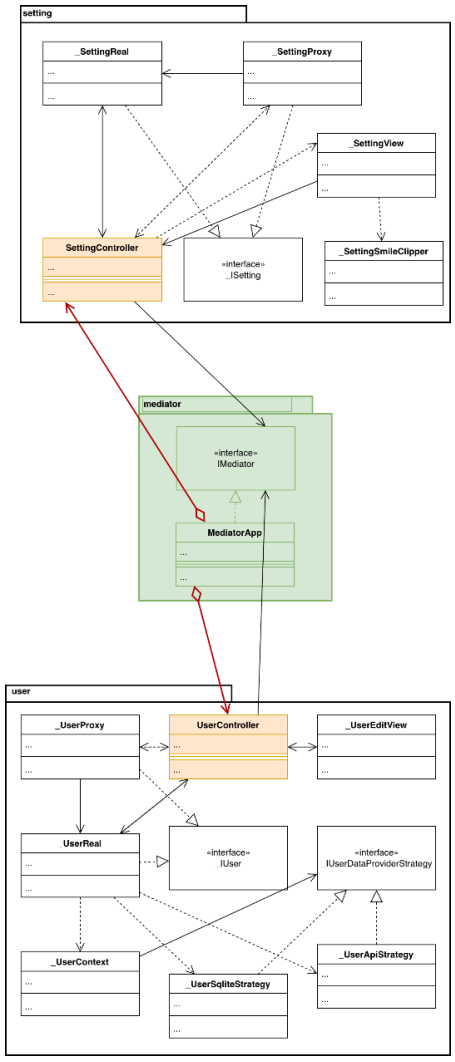


Fig. 3. UML diagram illustrating the mediator pattern implementation.

C. Implementation of Mediator Pattern

Fig. 3 illustrates the Mediator structure generated by the framework. In the evaluated generated architecture, the inter-module interaction represented in the structural comparison is coordinated through the Mediator component rather than through the direct Login–Register references observed in the legacy architecture.

The generated Mediator structure routes the evaluated inter-module interactions through a common coordination mechanism rather than through direct cross-module references. This organization reduces explicit relationships between the participating functional modules and may limit change propagation caused by direct references. However, the actual maintenance effort required to modify or extend the generated modules was not measured in the present study.

D. Implementation of Strategy Pattern

Fig. 4 illustrates the Strategy structure generated by the framework. The structure represents REST API and SQLite implementations through a common data-provider abstraction.

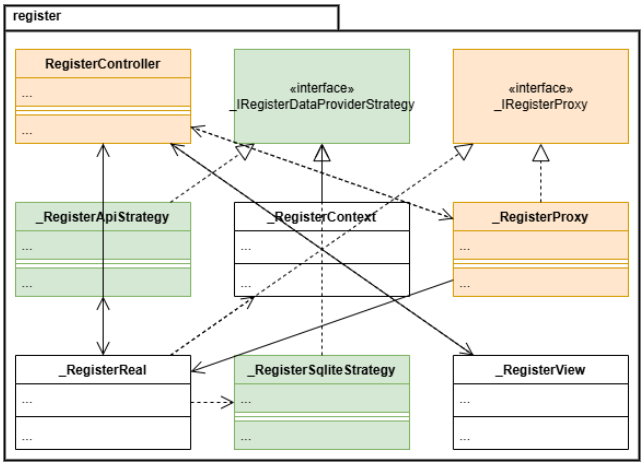


Fig. 4. UML diagram illustrating the strategy pattern implementation.

This organization separates provider-specific behavior from controller-facing logic and structurally supports provider substitution or extension. However, the experiment did not include a controlled change task measuring the files, source-code modifications, or developer effort required to introduce an additional provider. Therefore, the result is interpreted as structural support for extensibility rather than quantitative evidence of reduced maintenance effort.

E. Discussion

The evaluation provides complementary evidence concerning operational feasibility and architectural structure. Operationally, the proposed framework successfully generated, executed, and built Flutter applications from all three tested JSON specifications, covering one-, two-, and eight-module configurations. These results indicate that the generation mechanism can operate across the evaluated levels of functional complexity. Nevertheless, the limited number of specifications means that the findings demonstrate feasibility within the tested configuration space rather than general reliability across arbitrary Flutter applications.

The comparison between the legacy and generated architectures provides additional evidence regarding inter-module dependency management. In the evaluated Login–Register interaction, the legacy architecture contains two direct cross-module relationships, whereas the generated architecture replaces these direct relationships with communication through the Mediator component. Accordingly, the direct functional-module dependency count decreases from two to zero for this evaluated interaction. This result supports the intended role of the Mediator generation rule in reducing direct relationships among functional modules. However, the mediator remains an architectural dependency and may itself become a coordination point as the number and complexity of module interactions increase.

The Strategy implementation provides complementary structural support for variation in data-access behavior. REST API and SQLite implementations are represented through a common provider abstraction, allowing provider-specific behavior to remain separated from controller-facing logic. This demonstrates that the generator can systematically encode alternative data-access mechanisms within the generated architecture. Nevertheless, no controlled change-impact experiment was performed to quantify the effort required to introduce an additional provider. Therefore, the current evidence supports architectural extensibility at the structural level but does not establish a quantitative reduction in maintenance effort.

The results should also be interpreted within the limitations of the selected evaluation evidence. Direct functional-module relationships and UML-based architectural inspection capture observable aspects of dependency organization and generated structure, but they do not represent all dimensions of software quality. Maintainability, scalability, and testability are multidimensional properties that may also depend on source-code complexity, test coverage, runtime behavior, developer comprehension, defect history, and long-term evolution. Accordingly, the present study limits its conclusions to the architectural relationships and operational behavior directly supported by the evaluated artifacts.

A direct empirical benchmark against external Flutter code generators was not conducted because the reviewed approaches differ in functional scope, input representation, architectural objectives, and availability of comparable implementations. For this reason, existing approaches are compared at the capability level in the Related Work section, while the empirical architectural comparison uses the available legacy application as an internal baseline. A controlled benchmark using functionally equivalent application specifications and reproducible external generators remains an important direction for future work.

Overall, the findings indicate that the proposed architecture-aware generation approach can transform the evaluated JSON specifications into executable Flutter applications while applying predefined rules for module organization, mediator-based communication, and strategy-based data access. The evidence supports architectural conformance and controlled direct coupling in the evaluated cases, but broader claims regarding long-term maintainability, large-scale scalability, and developer productivity require additional longitudinal and comparative experiments.

V. Threats to Validity

The study does not report comprehensive quantitative source-code quality or architectural metrics, such as coupling measures, dependency-violation counts, cyclomatic complexity, generated lines of code, or class-level structural metrics. Accordingly, the reported structural evidence should be interpreted as targeted architectural inspection rather than as a comprehensive quantitative assessment of generated-code quality. Future evaluations should incorporate reproducibly defined source-code and architecture metrics alongside structural inspection. This study has several limitations that should be considered when interpreting the results.

A. Internal Validity

The generated applications depend on predefined templates and architectural generation rules. Consequently, defects or biases in these templates may influence the resulting architectural structures. To reduce this threat, the same generator, JSON configuration mechanism, software environment, and evaluation procedure were applied to all three experimental scenarios. The generated outputs and UML structures were also inspected to verify that the intended Mediator and Strategy relationships were present.

B. Construct Validity

The structural evaluation focuses on directly observable architectural relationships, including direct functional-module dependencies and the representation of Mediator- and Strategy-based structures. These observations capture selected aspects of architectural organization but do not directly measure broader software-quality attributes such as maintainability, scalability, testability, developer comprehension, or maintenance cost. Likewise, generation time, build time, and APK size are operational characteristics rather than direct measures of these quality attributes. Therefore, the findings are interpreted as evidence of operational feasibility and structural architectural conformance within the evaluated scenarios.

C. Conclusion Validity

The evaluation involved three application specifications, and each scenario was executed once. Consequently, the reported generation and build results are descriptive observations rather than statistical estimates of framework reliability or performance. No claim of statistical superiority over other code generators is made. The three specifications were selected to represent increasing functional complexity rather than to constitute a statistically representative sample of Flutter applications.

D. External Validity

The experiments were limited to Flutter applications generated from the supported JSON schema, predefined templates, and eight available functional modules. Applications requiring highly customized state management, platform-specific integrations, or architectural relationships outside the supported abstractions may require additional manual implementation. Therefore, the findings should be generalized

only to application structures that fall within the current framework configuration.

In practical terms, the current framework is most applicable to modular Flutter applications whose pages, routes, functional modules, and data-provider requirements can be represented by the supported JSON schema. Projects requiring highly customized architectural structures may require extensions to the schema or generation templates. Therefore, industrial adoption should also consider template customization and integration with manually maintained source code.

VI. Conclusion

This study proposed an architecture-aware Flutter code generation framework that transforms JSON-based application specifications into modular source code while systematically applying Clean Architecture, Mediator, and Strategy rules. The primary contribution is not the isolated use of these established architectural concepts, but their encoding as repeatable specification-to-code generation rules that are instantiated in the evaluated generated application structures.

The framework successfully generated, executed, and built all three evaluated application specifications containing one, two, and eight functional modules. In the structural comparison, the two direct Login–Register relationships observed in the legacy architecture were replaced by mediator-based communication in the generated architecture, resulting in no direct Login–Register relationship for the evaluated interaction. The generated Strategy structure also represented REST API and SQLite data providers through a common abstraction, demonstrating that the intended data-provider variation was incorporated into the generated architecture.

These findings should be interpreted within the scope of the experiment. The evaluation was limited to three predefined specifications, template-generated Flutter applications, selected architectural relationships, and single executions of each scenario. Therefore, the results do not constitute general proof of long-term maintainability, unlimited scalability, or superiority over other Flutter code generators.

Future work should evaluate larger industrial applications, perform repeated performance experiments, compare the framework with functionally equivalent and reproducible generators, conduct controlled maintenance and change-impact studies, and assess developer effort and comprehension. Additional work may also extend the JSON specification and generation templates to support further architectural and behavioral design patterns.

More broadly, architecture-aware code generation provides a practical direction for transforming architectural decisions from developer-dependent guidelines into explicit and repeatable software-construction rules.

Acknowledgment

The authors would like to thank SimHive Group (PT. Maulidan Teknologi Kreatif) for their cooperation and support during this research.

Declarations

Conflict of Interest: The authors declare that they have no competing interests that could have influenced the work reported in this manuscript.

Funding: This research received no financial support from any funding agency, institution, or commercial organization.

Data Availability: The data and supporting materials used in this study are available from the authors upon reasonable request.

Declaration of Generative AI Use: Generative AI tools were used only for language refinement and editorial assistance during manuscript preparation. All scientific content, analysis, interpretations, and conclusions remain the responsibility of the authors.

References

[1] S. A. Bhagat, “Review on Mobile Application Development Based on Flutter Platform,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 1, pp. 803–809, Jan. 2022, doi: 10.22214/ijraset.2022.39920.

[2] C. W. Butler and T. J. McCabe, “Cyclomatic Complexity-Based Encapsulation, Data Hiding, and Separation of Concerns,” Journal of Software Engineering and Applications, vol. 14, no. 01, pp. 44–66, 2021, doi: 10.4236/jsea.2021.141004.

[3] A. Khalilipour, M. Challenger, M. Onat, H. Gezgen, and G. Kardas, “Refactoring Legacy Software for Layer Separation,” International Journal of Software Engineering and Knowledge Engineering, vol. 31, no. 02, pp. 217–247, Feb. 2021, doi: 10.1142/S0218194021500066.

[4] T. Rachovski, S. Hadzhikoleva, E. Hadzhikolev, and A. Lengerov, “Using Clean Architecture Principles to Improve the Design and Implementation of the Mobiles Online Platform,” 2022, pp. 11–19. doi: 10.1007/978-981-16-7657-4_2.

[5] J. A. Santiago-Salazar and D. Rico-Bautista, “Clean Architecture: Impact on Performance and Maintainability of Native Android Projects,” 2024, pp. 82–90. doi: 10.1007/978-3-031-47372-2_8.

[6] M. B. Sinatria, Oman Komarudin, and Kamal Prihamdani, “Penerapan Clean Architecture Dalam Membangun Aplikasi Berbasis Mobile Dengan Framework Google Flutter,” Infotech Journal, vol. 9, no. 1, pp. 132–146, May 2023, doi: 10.31949/infotech.v9i1.5237.

[7] V. Sarcar, “Mediator Pattern,” in Java Design Patterns, Berkeley, CA: Apress, 2022, pp. 489–511. doi: 10.1007/978-1-4842-7971-7_23.

[8] P. Petrova and Z. Varbanov, “Implementation of the software design patterns mediator and CQRS with the library mediatr in .NET,” Mathematics, Computer Science and Education, vol. 7, no. 2, pp. 75–82, Dec. 2024, doi: 10.54664/URDG9808.

[9] V. Sarcar, “Strategy Pattern,” in Java Design Patterns, Berkeley, CA: Apress, 2022, pp. 455–468. doi: 10.1007/978-1-4842-7971-7_21.

[10] R. A. Wijayanto and RR. H. P. Sejati, “Implementing Flutter Clean Architecture for Mobile Tourism Application Development,” Int. J. Comput. Appl., vol. 185, no. 39, pp. 23–30, Nov. 2023, doi: 10.5120/ijca2023923197.

[11] A. Aljaloud and A. Razzaq, “An Innovative Metric-based Clustering Approach for Increased Scalability and Dependency Elimination in Monolithic Legacy Systems,” Engineering, Technology & Applied Science Research, vol. 13, no. 4, pp. 11375–11386, Aug. 2023, doi: 10.48084/etasr.6048.

[12] S. R. Idate, T. S. Rao, and D. J. Mali, “Context-Based Aspect-Oriented Requirement Engineering Model,” Engineering, Technology & Applied Science Research, vol. 13, no. 2, pp. 10460–10465, Apr. 2023, doi: 10.48084/etasr.5699.

[13] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, Dec. 2019, pp. 1115–1120. doi: 10.1109/CSCI49370.2019.00211.

[14] B. W. Stirgwolt, T. A. Mazzuchi, and S. Sarkani, "A model-based systems engineering approach for developing modular system architectures," Journal of Engineering Design, vol. 33, no. 2, pp. 95–119, Feb. 2022, doi: 10.1080/09544828.2021.1980203.

[15] R. S. Sangwan, R. L. Nord, and I. Ozkaya, "Architectural Dependency Analysis: Addressing the Elephant in the Room," Computer (Long. Beach. Calif)., vol. 54, no. 3, pp. 73–78, Mar. 2021, doi: 10.1109/MC.2021.3050538.

[16] P. Sun, D.-K. Kim, H. Ming, and L. Lu, "Measuring Impact of Dependency Injection on Software Maintainability," Computers, vol. 11, no. 9, p. 141, Sep. 2022, doi: 10.3390/computers11090141.

[17] K. Lano, H. Haughton, Z. Yuan, and H. Alfraihi, "Agile model-driven re-engineering," Innov. Syst. Softw. Eng., vol. 20, no. 4, pp. 559–584, Dec. 2024, doi: 10.1007/s11334-024-00568-z.

[18] L. Jiménez-Navajas, R. Pérez-Castillo, and M. Piattini, "Code generation for classical-quantum software systems modeled in UML," Softw. Syst. Model., vol. 24, no. 3, pp. 795–821, Jun. 2025, doi: 10.1007/s10270-024-01259-w.

[19] I. B. Tiali, Z. Aarab, A. Lyazidi, and M. D. Rahmani, "Model-Driven Transformation of Business Processes into Blockchain Smart Contracts," International Journal of Advanced Computer Science and Applications, vol. 16, no. 12, 2025, doi: 10.14569/IJACSA.2025.01612120.

[20] C. P. A. Yataco, M. A. L. Salinas, and J. A. H. Quispe, "Mobile Application Based on Clean Architecture for Sustainable Tourism Recommendation Using BERT," International Journal of Advanced Computer Science and Applications, vol. 16, no. 11, 2025, doi: 10.14569/IJACSA.2025.0161199.

[21] S. Abufakher, "Empirical Quantitive Investigation of the Effect of GoF Design Patterns on the Quality of Software Systems," International Journal of Advanced Computer Science and Applications, vol. 17, no. 3, 2026, doi: 10.14569/IJACSA.2026.0170371.

A Unified Agent-Based Simulation Framework for Epidemic and Wildfire Propagation in Complex Adaptive Systems

Natasha Stojkovikj, Aleksandra Stojanova Ilievska, Limonka Koceva Lazarova, Daniel Eftimov, Irena Eftimova
Faculty of Computer Science, Goce Delcev University, Stip, North Macedonia

***Abstract*—Spreading phenomena in complex adaptive systems often exhibit similar dynamic characteristics despite originating from different application domains. This study investigates heterogeneous spreading processes using a common agent-based modeling methodology implemented in the AnyLogic simulation platform. Two representative models were developed: influenza transmission within a university population and wildfire propagation in a forest environment. Both models employ statechart-based agents, stochastic state transitions, and local interactions while preserving domain-specific characteristics. The influenza model incorporates mask usage, vaccination, and isolation as intervention strategies. Simulation results show that the investigated intervention scenarios reduced epidemic intensity compared with the baseline under the specific parameter settings considered, with peak infectious populations ranging from 44 to 70 individuals compared with 104 in the baseline scenario. The wildfire model investigates the influence of vegetation density, wind conditions, and burning duration, with burned trees ranging from 1 to 1,035 depending on environmental conditions. Comparative analysis demonstrated that both systems exhibit marked changes in propagation dynamics across the investigated scenarios, together with strong dependence on local interactions and stochastic processes. The successful application of a common agent-based modeling methodology across two distinct spreading phenomena highlights its consistency and transferability while supporting decision-making in public health and environmental risk management.**

***Keywords*—Agent-based modeling; AnyLogic; complex adaptive systems; influenza transmission; wildfire propagation**

I. Introduction

Many natural and social systems are characterized by processes that spread through interactions among neighboring entities. Examples include infectious diseases, wildfires, information diffusion, and cascading failures in infrastructure systems. Although these phenomena belong to different scientific domains, they often exhibit similar propagation patterns influenced by local interactions, stochastic effects, threshold behavior, and nonlinear dynamics. Understanding these processes is important for improving prevention strategies, supporting decision-making, and reducing potential societal, environmental, and economic impacts. Despite differences in their physical mechanisms, many spreading phenomena exhibit comparable emergent behaviors that can be investigated using similar agent-based modeling concepts.

Agent-based modeling (ABM) has been widely used for studying complex systems because it allows individual entities, their behavior, and interactions to be represented explicitly. In contrast to aggregate mathematical approaches, agent-based simulations enable researchers to investigate how relatively simple local rules may lead to complex system-level behavior. This capability has contributed to the increasing use of ABM in epidemiology, ecology, transportation systems, disaster management, and social sciences. In addition, modern simulation environments such as AnyLogic provide flexible mechanisms for implementing state-based agents, spatial interactions, and stochastic processes within a unified modeling framework. Such flexibility enables heterogeneous spreading phenomena to be investigated using a common modeling methodology, facilitating cross-domain comparisons while improving the transparency, interpretability, and reproducibility of simulation studies.

Seasonal influenza remains one of the most common infectious diseases worldwide and continues to place a considerable burden on healthcare systems and society. Educational institutions represent environments in which disease transmission can occur rapidly due to frequent interpersonal contacts, shared facilities, and high population densities. Investigating the dynamics of influenza outbreaks and evaluating the effects of protective measures are therefore important research topics in public health and epidemic modeling.

Wildfires represent another type of spreading phenomenon that has received substantial attention in recent years. Increasing temperatures, prolonged drought periods, and changing climatic conditions have contributed to more frequent and severe wildfire events in many regions of the world. Fire propagation depends on multiple factors, including vegetation density, wind conditions, burning duration, and local neighborhood interactions. Understanding how these factors influence wildfire dynamics is essential for improving prevention strategies, optimizing resource allocation, and supporting emergency response planning.

Most existing studies investigate epidemic transmission and wildfire propagation as separate research problems, primarily focusing on domain-specific models and applications. Consequently, relatively limited attention has been devoted to developing unified agent-based modeling frameworks that enable the systematic investigation of heterogeneous spreading phenomena using consistent modeling principles and comparable simulation structures. Although these phenomena share several fundamental propagation characteristics, including

local interactions, stochastic state transitions, and emergent behavior, their common modeling potential has received comparatively little attention. A cross-domain perspective may therefore provide a broader understanding of generic propagation mechanisms in complex adaptive systems while demonstrating the transferability of common agent-based modeling principles across different application domains.

Therefore, the objective of this study is to investigate whether different spreading phenomena can be consistently analyzed using a common agent-based modeling methodology implemented in the AnyLogic simulation environment. To achieve this objective, two agent-based models were developed. The first model represents influenza transmission within a higher education institution, while the second simulates wildfire propagation in a forest environment. Both models employ statechart-based agents, discrete spatial representations, and stochastic local interactions. Their comparative analysis enables the identification of common propagation characteristics while illustrating the applicability of agent-based simulation for studying heterogeneous spreading processes in complex adaptive systems.

The main contributions of this study are threefold. First, two comparable agent-based simulation models representing biological and environmental spreading phenomena are developed using a common statechart-based modeling methodology implemented in the AnyLogic platform. Second, the proposed models are comparatively evaluated under multiple intervention and environmental scenarios to investigate similarities and differences in propagation dynamics. Third, the study identifies common characteristics of spreading processes across heterogeneous complex adaptive systems, demonstrating the generalizability of agent-based simulation for cross-domain modeling of spreading phenomena.

The remainder of the study is organized as follows. Section II presents an overview of related studies in epidemic simulation, wildfire modeling, and agent-based methodologies. Section III introduces the theoretical concepts underlying the proposed models. Section IV describes the development and implementation of the influenza and wildfire simulations. Experimental results and comparative analyses are discussed in Section V, while Section VI summarizes the main findings and outlines potential directions for future research.

II. Related Work

Agent-based modeling (ABM) has become one of the most widely adopted approaches for studying complex adaptive systems characterized by heterogeneous entities, decentralized interactions, and emergent behavior. Unlike aggregate mathematical models, ABM explicitly represents individual agents, their behavioral rules, and interactions within a dynamic environment, allowing researchers to investigate how local decisions and stochastic processes collectively shape system-level dynamics. The applicability of agent-based techniques has been demonstrated in numerous scientific disciplines, including epidemiology, social sciences, transportation systems, ecology, and disaster management [1]–[4]. The development of multimethod simulation environments has additionally facilitated the implementation of ABM by enabling the integration of agent-based, discrete-event, and system dynamics paradigms within a common modeling framework [4].

The spread of infectious diseases has traditionally been investigated using compartmental epidemiological models. The classical SIR framework remains one of the most important theoretical foundations for epidemic modeling [5], while subsequent studies extended these concepts and demonstrated their usefulness for understanding disease dynamics and evaluating intervention strategies [6]. However, compartmental models generally assume homogeneous populations and often neglect heterogeneity in contact structures, mobility patterns, and individual behavior. To overcome these limitations, network-based and agent-based approaches have gained increasing attention [7]–[9].

Numerous studies have investigated influenza outbreaks and pandemic mitigation measures. Strategies for containing pandemic influenza at its source were investigated in [10], while targeted layered containment approaches were modeled in [11]. The role of herd immunity in infectious disease control was examined in [12]. School closure as a mitigation strategy was investigated in [13] and further evaluated in the context of the 2009 H1N1 pandemic in [14]. Factors affecting the controllability of infectious disease outbreaks were analyzed in [15], whereas a stochastic agent-based approach for epidemic simulation was developed in [16]. More recent research has expanded the methodological scope of epidemic modeling. An agent-based model incorporating non-pharmaceutical interventions and contact tracing was presented in [17]. A broader analysis of agent-based epidemic modeling approaches, applications, and future directions was provided in [18], while epidemic processes in complex networks were examined in [19].

Complex network theory has also played an important role in epidemic research. The influence of social network structures, connectivity patterns, and contact heterogeneity on disease transmission has been widely investigated [7], [20], [21]. These studies demonstrate that epidemic propagation is strongly influenced by local interactions among individuals as well as by the topology of the underlying contact network.

Wildfire propagation has been investigated through empirical, physical, and simulation-based approaches. Early studies demonstrated that simple local ignition mechanisms may generate self-organized critical behavior in forest-fire systems [22], [23]. Percolation theory has frequently been employed to explain the existence of critical vegetation density thresholds that significantly influence fire spread [24]. Cellular automata have also proven effective in modeling wildfire behavior, particularly when accounting for neighborhood interactions and environmental conditions [25]. More recent wildfire studies address different aspects of model realism. An efficient wildfire propagation approach using irregular grids was proposed in [26]. The effects of input-data variation and parameter adjustment were investigated in [27], while wildfire spread models addressing spotting phenomena were systematically reviewed in [28]. Together, these studies highlight complementary directions for improving spatial representation, parameterization, and the representation of complex fire-spread mechanisms. However, comparatively little attention has been devoted to investigating whether these modeling principles can

be generalized and consistently applied across different categories of spreading phenomena.

Although epidemic transmission and wildfire propagation are generally investigated within different scientific disciplines, both represent spreading phenomena governed by local interactions, stochastic state transitions, neighborhood effects, and nonlinear emergent dynamics. Consequently, both systems can be investigated using similar agent-based modeling concepts despite their fundamentally different physical mechanisms. Recent studies have additionally demonstrated the applicability of AnyLogic for modeling epidemic processes, educational environments, and other complex adaptive systems [29]–[32]. Nevertheless, these studies primarily focus on individual application domains rather than proposing a unified methodological framework for investigating heterogeneous spreading phenomena.

Building upon these observations, the present study adopts a common agent-based modeling methodology, implemented in the AnyLogic simulation environment, to investigate two different spreading phenomena. Rather than treating epidemic transmission and wildfire propagation as isolated case studies, the proposed approach explores their common propagation characteristics using comparable modeling principles, thereby providing a cross-domain perspective on the analysis of spreading dynamics in complex adaptive systems.

III. Theoretical Background

Agent-based modeling (ABM) is a bottom-up simulation approach in which autonomous entities interact within a predefined environment according to specific behavioral rules. In contrast to aggregate mathematical models, agent-based simulations describe the state and behavior of individual entities explicitly, making it possible to investigate how local interactions and stochastic events contribute to the emergence of system-level dynamics [1]–[4]. The ability to incorporate heterogeneous populations, spatial relationships, and adaptive behavior has led to the widespread use of ABM in epidemiology, ecology, transportation systems, and disaster management.

The behavior of agents is commonly represented through statecharts, which define possible states and transitions among them. Statecharts provide a convenient mechanism for modeling event-driven and time-dependent processes, allowing agents to modify their states according to probabilistic rules, interactions with neighboring entities, or changes in environmental conditions [4]. This approach is particularly useful when studying spreading processes, where the state of one agent may directly affect the evolution of nearby agents.

Epidemic spread is traditionally described using compartmental models in which individuals transition between epidemiological states such as susceptible, infected, and recovered [5], [6]. Although these models provide valuable insights into disease dynamics, they generally assume homogeneous mixing within the population and often neglect variations in contact patterns, mobility, and individual behavior. Agent-based approaches address these limitations by explicitly representing social interactions and behavioral heterogeneity, enabling a more realistic description of disease transmission [7]–[9]. Recent studies have further demonstrated the applicability of agent-based frameworks for incorporating intervention strategies, contact tracing mechanisms, and non-pharmaceutical mitigation measures [16]–[19].

Wildfire propagation can also be considered a spreading process governed by local interactions. In many simulation studies, trees are represented as discrete entities occupying cells within a spatial grid and may transition between states such as intact, burning, and burned. The likelihood that a tree ignites is influenced by several factors, including vegetation density, burning duration, wind conditions, and the characteristics of neighboring cells [21]–[27]. Previous investigations have shown that wildfire dynamics exhibit properties associated with self-organized criticality and percolation phenomena, where relatively small changes in vegetation density may substantially alter the probability of large-scale fire events [22]–[24].

Although epidemic transmission and wildfire propagation are generally investigated within different scientific disciplines, both can be formulated as spreading processes driven by interactions among autonomous agents evolving. From a simulation perspective, both systems share common modeling elements, including discrete agent states, stochastic state transitions, local neighborhood interactions, spatial environments, and scenario-dependent parameters. Consequently, similar agent-based modeling principles can be employed to investigate different spreading phenomena while maintaining a consistent simulation methodology.

Based on these common modeling principles, both simulation models presented in this study were implemented using the AnyLogic simulation platform [33], which provides integrated support for statechart-based agents, stochastic processes, spatial interactions, and scenario management [4]. Employing the same simulation environment and modeling methodology enables a consistent comparative investigation of biological and environmental spreading phenomena while preserving the specific characteristics of each application domain. This unified methodological perspective provides the basis for the comparative analyses presented in the following sections.

A. Common Agent-Based Modeling Methodology

The proposed models follow the same agent-based modeling methodology despite representing different application domains. In both models, the system is composed of autonomous agents situated within a discrete spatial environment. Each agent occupies one state at a given time and changes its state through statechart-based transitions triggered by local interactions, stochastic processes, or predefined simulation rules. The environment defines spatial relationships among neighboring agents, while scenario-specific parameters control interaction probabilities and propagation dynamics. Simulation outputs are obtained by observing the collective behavior emerging from numerous local interactions over time. The adoption of a common modeling methodology facilitates consistent comparative analysis, improves the interpretability of simulation outcomes, and demonstrates the transferability of agent-based modeling principles across different categories of spreading phenomena.

In this study, the term unified refers to the use of a common modeling methodology rather than to a shared mathematical formulation or identical parameter set. Both models follow the same sequence of agent specification, statechart design, local interaction modeling, scenario configuration, simulation execution, and comparative analysis, while their transition mechanisms and parameters remain domain-specific.

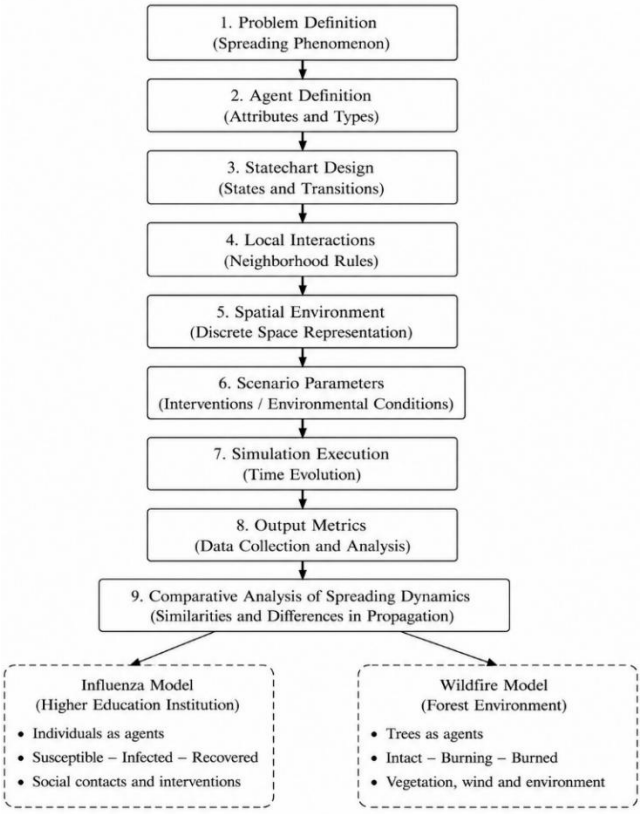


Fig. 1. General workflow of the common agent-based modeling methodology employed in both simulation models.

Fig. 1 summarizes the common agent-based modeling workflow adopted in this study. Although the influenza and wildfire models represent different application domains, both follow the same sequence of problem definition, agent specification, statechart design, local interaction modeling, scenario configuration, simulation execution, and comparative analysis. This workflow defines a reusable modeling procedure in which a new spreading domain can be represented by specifying its agents and states, defining domain-specific transition and local interaction rules, configuring relevant scenario parameters, executing the simulation, and analyzing the resulting system-level behavior. The following sections describe the implementation details of each application while maintaining this common methodological structure.

Although the influenza and wildfire models represent different application domains, they share several common modeling principles that ensure methodological consistency throughout the simulation process. Table I summarizes the principal modeling elements adopted in both agent-based models and highlights the similarities and domain-specific characteristics of each application.

TABLE I. COMPARISON OF THE COMMON MODELING ELEMENTS ADOPTED IN THE INFLUENZA AND WILDFIRE AGENT-BASED SIMULATION MODELS

Modeling Element	Influenza Model	Wildfire Model
Agents	**Students**	**Trees**
State representation	Susceptible, Exposed, Infectious, Recovered	Unburned, Burning, Burned
Environment	Higher education institution	Forest environment
Local interactions	Person-to-person contacts	Neighboring trees
State transitions	Infection, incubation, recovery	Ignition, burning, burnout
Scenario parameters	Vaccination, masks, isolation	Density, wind, burning duration
Output metrics	Epidemic dynamics	Fire propagation dynamics

As shown in Table I, both simulation models follow the same methodological structure despite representing different spreading phenomena. The primary differences relate to the definition of agents, state transitions, and domain-specific parameters, whereas the underlying agent-based modeling principles, statechart implementation, local interactions, and scenario-based experimentation remain consistent. This methodological consistency forms the basis for the detailed model descriptions presented in the following sections.

IV. METHODOLOGY AND SIMULATION SETUP

A. Influenza Transmission Model

Following the common agent-based modeling methodology described in Section III-A, the first application investigates influenza transmission within a higher education institution. The model was implemented in the AnyLogic simulation environment, where students are represented as autonomous agents interacting within a shared environment and transitioning among epidemiological states according to predefined behavioral rules (Fig. 2).

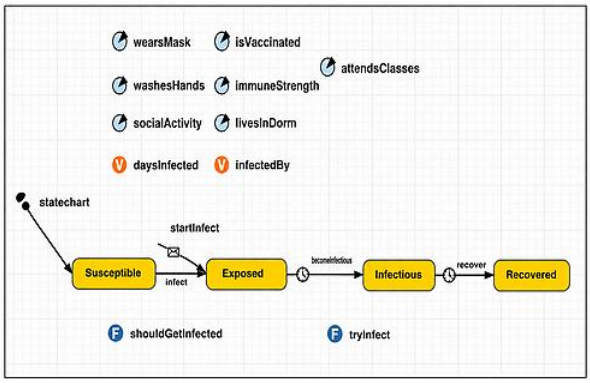


Fig. 2. Statechart of the influenza transmission model.

The implemented statechart consists of four epidemiological states, namely Susceptible, Exposed, Infectious, and Recovered, allowing the representation of an incubation period prior to the onset of infectiousness. State transitions are governed by probabilistic mechanisms describing disease transmission, progression, and recovery. The statechart implementation follows the common modeling workflow presented in Fig. 1,

where individual agent behavior emerges through state transitions triggered by local interactions and stochastic events.

At the beginning of each simulation run, a predefined percentage of students is initialized as infectious, whereas the remaining population is considered susceptible. Disease transmission occurs through interactions between susceptible and infectious individuals. The probability of infection depends on a transmission coefficient that can be adjusted to represent different outbreak conditions. Exposed individuals remain in the incubation state for a specified period before becoming infectious. After the infectious period, agents transition to the recovered state, where temporary immunity is assumed.

The simulation population consisted of 200 students, with three individuals initially infected. The incubation and recovery periods were configured within the ranges of 1–7 and 2–14 days, respectively. Disease transmission was governed by probabilistic interactions between susceptible and infectious agents, with the transmission coefficient controlling the likelihood of infection. The intervention scenarios considered 80% mask compliance, 75% vaccination coverage, and a 70% isolation probability. In the isolation scenario, students maintained an average of twelve daily contacts and could enter isolation only after at least two days in the infectious state.

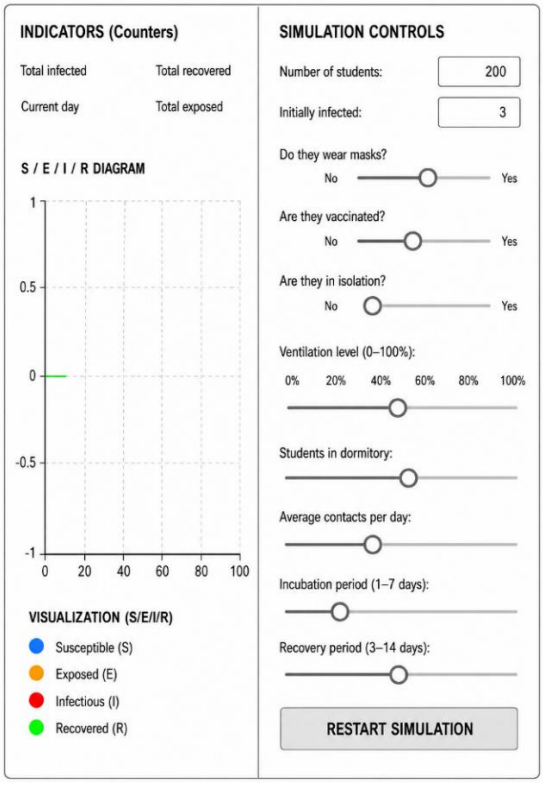


Fig. 3. Simulation environment of the influenza model.

To evaluate the influence of mitigation strategies on epidemic dynamics, several intervention mechanisms were incorporated into the model, including vaccination, mask usage, self-isolation, and reductions in contact frequency. These measures allow the investigation of their impact on epidemic dynamics and facilitate comparative analyses under different mitigation strategies. The simulation environment and the main model parameters used to configure the influenza scenarios are shown in Fig. 3.

B. Wildfire Propagation Model

The second application follows the same agent-based modeling methodology and investigates wildfire propagation within a forest environment. Similar to the influenza model, the wildfire simulation was implemented using statechart-based agents in the AnyLogic platform. Individual trees occupy cells of a two-dimensional spatial grid and interact locally with neighboring trees according to predefined propagation rules (Fig. 4).

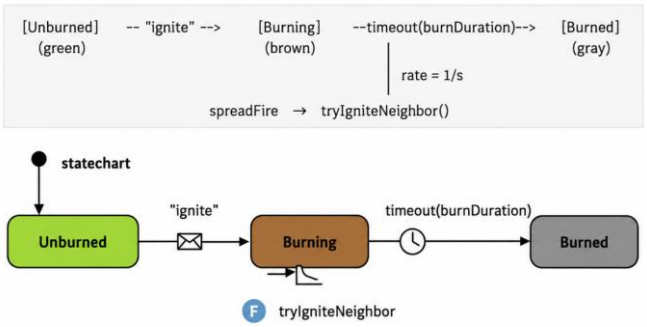


Fig. 4. Statechart of the wildfire propagation model.

Each tree may exist in one of three states: Unburned, Burning, and Burned. State transitions are implemented through an AnyLogic statechart. The wildfire statechart preserves the same state-based modeling philosophy adopted in the influenza model while incorporating domain-specific transition rules associated with fire propagation. A tree enters the burning state upon receiving an ignition message from a neighboring burning tree and remains active for a predefined burning duration before permanently transitioning to the burned state.

Fire propagation occurs locally through interactions among neighboring trees. The ignition probability depends on several factors, including forest density, burning duration, wind speed, and wind direction. Wind effects are incorporated by increasing ignition probabilities in the prevailing wind direction while simultaneously reducing propagation probabilities in the opposite direction.

Forest density, wind speed, wind direction, ignition probability, and burning duration were defined through adjustable simulation parameters accessible via the graphical user interface. These settings enabled the evaluation of wildfire propagation under different environmental conditions. The use of adjustable parameters enabled systematic scenario-based experimentation while maintaining consistency with the common simulation methodology described in Section III-A.

The wildfire environment was represented by a 50 × 50 spatial grid, with a base ignition probability of 0.3 in the illustrated configuration. The experimental scenarios considered forest densities of 0.3, 0.7, and 0.9, wind speeds of 0, 5, and 15 m/s, and burning durations of 2 and 5 simulation time units. Ignition depended on local interactions between neighboring trees, with wind direction modifying the likelihood of propagation along different directions. The complete scenario

configurations and corresponding simulation outcomes are summarized in Table III.

The initial forest configuration and the simulation environment used to configure the wildfire scenarios are shown in Fig. 5.

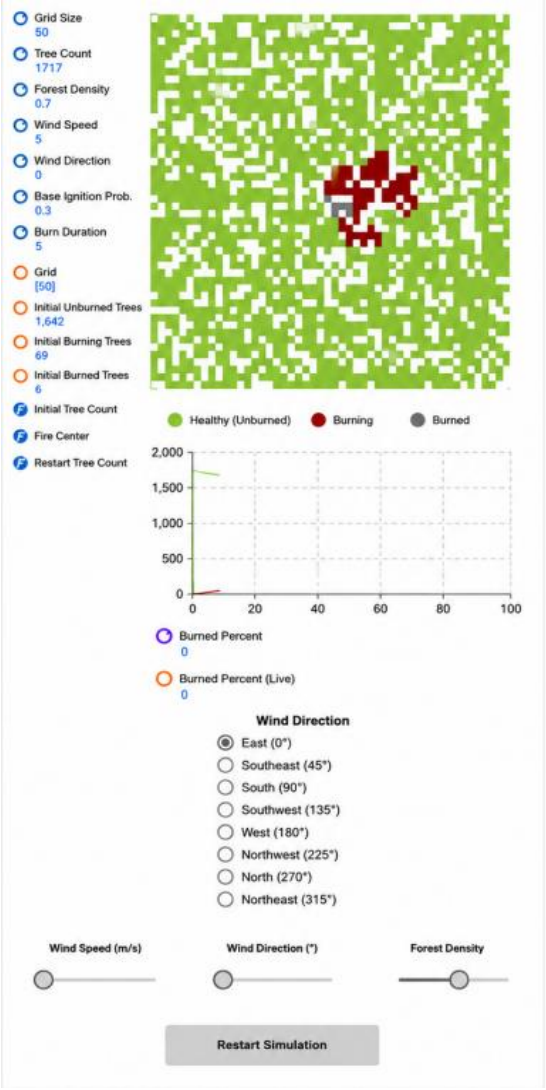


Fig. 5. Initial forest configuration and wildfire simulation environment.

V. Results and Discussion

The results reported in this section correspond to representative simulation runs for each investigated scenario. Because the models include stochastic state transitions, individual runs may produce variations in peak magnitude, timing, and outcomes. Therefore, the reported values should be interpreted as illustrative scenario outcomes rather than statistical estimates of expected system behavior.

A. Epidemic Dynamics Under Different Intervention Strategies

The proposed agent-based influenza transmission model was evaluated under four intervention scenarios designed to investigate the impact of various mitigation measures on epidemic progression within a university environment. The investigated scenarios included a baseline case without interventions, increased mask usage, vaccination, and isolation of infected individuals.

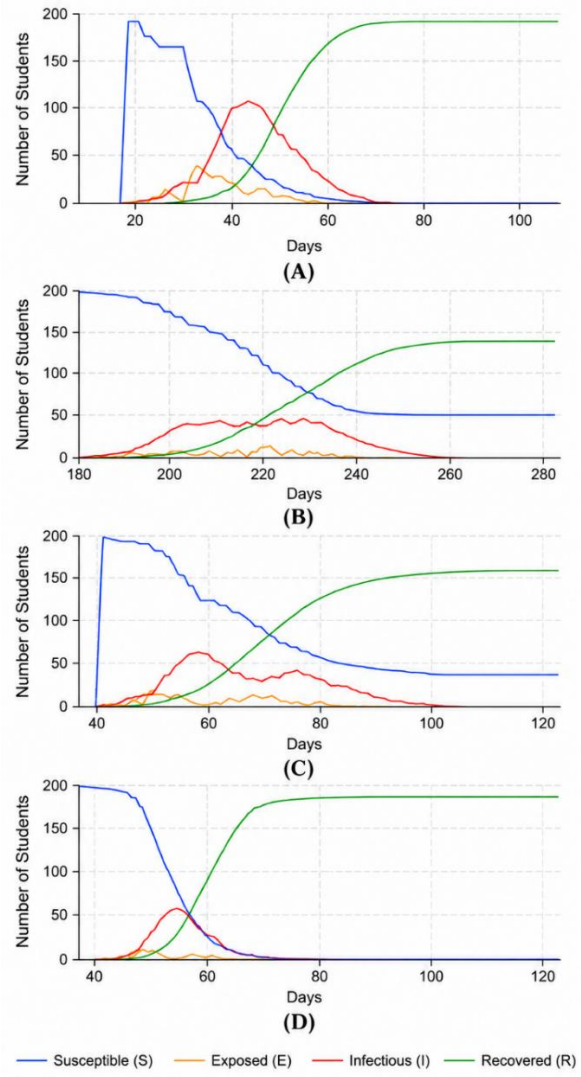


Fig. 6. Temporal evolution of Susceptible (S), Exposed (E), Infectious (I), and Recovered (R) Populations under four intervention scenarios: (A) Baseline scenario without interventions, (B) Mask usage, (C) Vaccination, and (D) Isolation of infected students.

Fig. 6 illustrates the temporal evolution of the SEIR compartments under the investigated intervention strategies. Significant differences in epidemic progression can be observed depending on the applied mitigation measure. The investigated scenarios were selected to evaluate the capability of the common agent-based modeling methodology to represent different intervention strategies using identical modeling principles while varying only domain-specific parameters.

The baseline scenario (Fig. 6A) represents disease transmission without preventive interventions. The epidemic spreads rapidly through the student population, reaching a maximum of approximately 104 simultaneously infectious students around day 15. By the end of the simulation, only six individuals remain susceptible, while 194 students recover from

the disease. These results indicate that approximately 97% of the population experiences infection in the absence of control measures.

The second scenario (Fig. 6B) investigates the effect of increased mask usage, assuming an 80% compliance rate among students. Compared with the baseline scenario, the epidemic peak is considerably reduced to approximately 45 infectious individuals and delayed until around day 50. The number of susceptible individuals at the end of the simulation increases to 52, while 148 students recover. The delayed epidemic peak and flatter epidemic curve suggest that mask usage effectively decreases transmission opportunities and slows disease propagation.

Vaccination was examined in the third scenario (Fig. 6C), assuming vaccination coverage of 75% of the student population. The number of infectious individuals reaches a maximum of approximately 44 students around day 15. Although vaccination substantially suppresses disease transmission, a smaller secondary increase in infectious cases can still be observed. At the end of the simulation, 34 students remain susceptible, while 166 recover from infection. These findings indicate that vaccination significantly improves epidemic control but may not eliminate disease spread among residual susceptible subgroups.

The fourth scenario (Fig. 6D) evaluates the impact of isolation as the sole intervention strategy. The isolation probability was set to 70%, whereas masks and vaccination remained disabled, and students maintained an average of twelve daily contacts. In contrast to masks and vaccination, isolation acts after infection has already occurred. Students may isolate only after being infectious for at least two days, allowing disease transmission during the initial infectious stage. The simulation results demonstrate a peak of approximately 70 infectious students around day 25. By the end of the simulation, 180 students recover, and only 20 remain susceptible. Under this scenario configuration, isolation reduced epidemic spread relative to the baseline, resulting in a peak of approximately 70 infectious students and an attack rate of 90%.

To facilitate comparison among the investigated intervention strategies, the principal epidemiological indicators extracted from the simulations are summarized in Table II.

As shown in Table II, all investigated intervention scenarios produced lower epidemic peaks and attack rates than the baseline scenario under the specific parameter settings considered. The mask scenario (80% compliance) resulted in a peak of 45 infectious individuals and an attack rate of 74%, while the vaccination scenario (75% coverage) produced a peak of 44 infectious individuals and an attack rate of 83%. The isolation scenario (70% isolation probability) resulted in a peak of 70 infectious individuals and an attack rate of 90%. Because these scenarios represent different intervention mechanisms and use different coverage or compliance levels, the results should not be interpreted as a normalized ranking of intervention effectiveness. Instead, they illustrate how the simulated epidemic dynamics changed under the specific intervention configurations investigated in this study.

TABLE II. COMPARATIVE EPIDEMIOLOGICAL INDICATORS OBTAINED FROM FOUR INTERVENTION SCENARIOS

	Intervention Strategy	Peak Infectious	Peak Day	Susceptible at End	Recovered at End	Attack Rate
A	Baseline	104	15	6	194	97.0 %
B	Masks (80%)	45	50	52	148	74.0 %
C	Vaccination (75%)	44	15	34	166	83.0 %
D	Isolation (70%)	70	25	20	180	90.0 %

B. *Wildfire Propagation Analysis*

To evaluate the proposed wildfire propagation model, four representative simulation scenarios were investigated by varying forest density, wind conditions, and burning duration. The resulting fire dynamics are illustrated in Fig. 7.

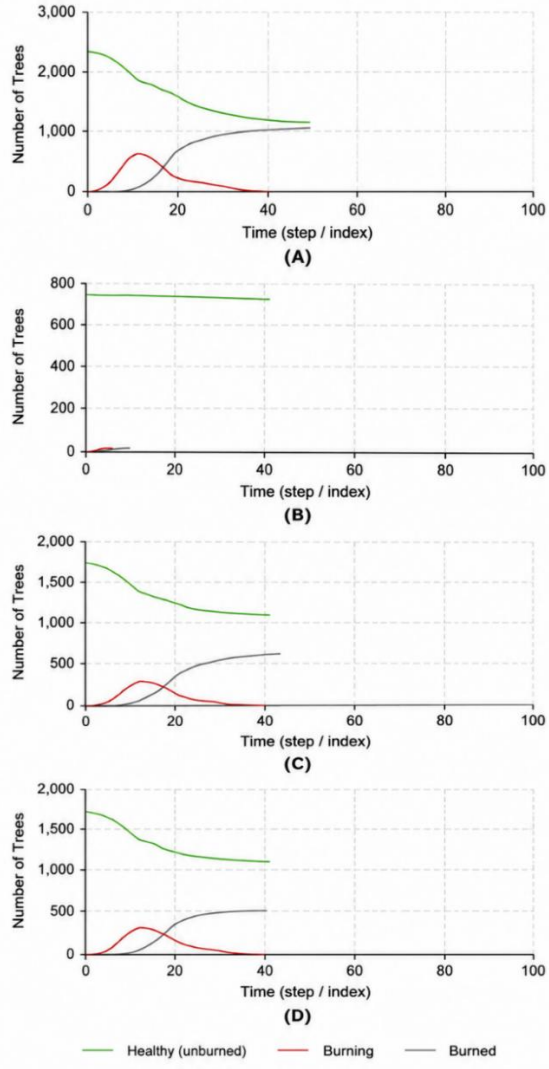


Fig. 7. Temporal evolution of healthy, burning, and burned trees under four representative wildfire propagation scenarios: (A) Dense forest without wind, (B) Sparse forest without wind, (C) Medium-density forest with strong wind, and (D) Reduced burning duration.

The investigated scenarios were selected to illustrate wildfire behavior under different combinations of the principal environmental parameters considered in the model. Consistent with the influenza case study, all wildfire scenarios preserve the same agent-based modeling methodology while using different application-specific environmental parameter settings. The scenarios were not designed as a factorial experiment to isolate the independent effect of each parameter. A summary of the considered configurations is presented in Table III.

TABLE III. COMPARATIVE SUMMARY OF WILDFIRE SIMULATION SCENARIOS

	Forest Density	Wind Speed	Burn Duration	Burned Trees	Peak Burning	Duration (steps)	Pattern
W1	0.9	0	5	1035	501	36	Circular
W2	0.3	0	5	1	3	6	Fragmented
W3	0.7	15 m/s East	5	617	239	46	Eastward
W4	0.7	5 m/s East	2	493	85	24	Reduced

The first scenario (Fig. 7A) represents a dense forest configuration with a vegetation density of 0.9 and no wind influence. Under these conditions, the fire spreads almost uniformly in all directions, producing a nearly circular propagation pattern. The fire spread rapidly through the highly connected vegetation, reaching a peak of approximately 501 simultaneously burning trees around simulation step 8. By the end of the simulation, 1,035 trees had burned, while 1,232 trees remained unaffected, indicating extensive fire damage under dense vegetation conditions. Since neighboring cells are highly connected, the fire rarely encounters natural barriers that would prevent further expansion.

The second scenario (Fig. 7B) investigates a sparse forest characterized by a density of only 0.3. In contrast to the previous case, numerous empty cells interrupt connectivity among trees, frequently leading to early fire extinction. The fire propagation remained highly localized. The maximum number of simultaneously burning trees did not exceed three, and only a single tree became permanently burned by the end of the simulation. These findings indicate that sparse vegetation acts as an effective natural barrier preventing large-scale fire spread.

The third scenario (Fig. 7C) considers a medium forest density of 0.7 and an eastward wind of 15 m/s. Under this parameter combination, the fire developed an elongated pattern directed along the wind axis. The simulation lasted 46 steps and resulted in 617 burned trees, while the maximum number of simultaneously burning trees reached approximately 239 around simulation step 13. Because this scenario differs from the no-wind scenarios in more than one parameter setting, the observed propagation pattern should be interpreted as specific to the investigated configuration rather than as an isolated estimate of the wind effect.

The final scenario (Fig. 7D) considers a forest density of 0.7, an eastward wind speed of 5 m/s, and a burning duration of 2 simulation time units. Under this parameter combination, 493 trees were burned, the maximum number of simultaneously burning trees was 85, and the propagation duration was 24 simulation steps. In comparison, Scenario W3 resulted in 617 burned trees, a peak of 239 simultaneously burning trees, and a duration of 46 simulation steps. Because W3 and W4 differ in both wind speed and burning duration, the observed differences cannot be attributed to burning duration alone.

Overall, the investigated scenarios produced substantial differences in fire extent, peak fire intensity, and propagation duration under different combinations of vegetation density, wind conditions, and burning duration. Because multiple parameters vary across the scenarios, these results describe scenario-specific propagation patterns rather than independent effects of individual environmental factors.

C. Comparative Discussion

The two developed simulation models demonstrate that different spreading phenomena can be investigated using a common agent-based modeling methodology while preserving their domain-specific characteristics. Although the biological and environmental processes considered in this study differ substantially in their underlying mechanisms, both follow identical modeling principles regarding autonomous agents, stochastic state transitions, local interactions, and scenario-based experimentation.

The influenza model considers students as individual agents interacting through social contacts and transitioning among susceptible, exposed, infectious, and recovered states. In contrast, the wildfire model represents trees as stationary agents whose states evolve from unburned to burning and eventually to burned. Despite these differences, both models employ statechart-based agent representations and incorporate probabilistic mechanisms governing state transitions.

To emphasize similarities and differences between the two simulation approaches, a comparative overview is presented in Table IV.

TABLE IV. COMPARATIVE CHARACTERISTICS OF THE DEVELOPED AGENT-BASED MODELS

Characteristic	Influenza Model	Wildfire Model
Simulation Platform	AnyLogic	AnyLogic
Modeling Approach	Agent-Based Modeling	Agent-Based Modeling
Agent Type	Student	Tree
State Representation	SEIR	Unburned–Burning–Burned
Statechart Implementation	Yes	Yes
Stochastic Components	Yes	Yes
Spatial Representation	Social interaction network	Two-dimensional grid
Main Influencing Factors	Masks, vaccination, isolation	Forest density, wind, burning duration
Main Performance Indicator	Peak infectious population	Percentage of burned trees
Intervention Strategies	Yes	No
Primary Objective	Epidemic mitigation	Wildfire propagation assessment

Table IV further demonstrates that, despite substantial differences in the modeled systems, the two simulation models preserve a consistent methodological structure. The principal distinctions arise from application-specific agents, states, and parameters, whereas the underlying simulation methodology remains identical.

The results obtained from both models indicate substantial differences in propagation dynamics across the investigated scenario settings. Similar observations have been reported in recent wildfire propagation studies employing calibrated simulation environments and irregular computational grids [26], [27]. In the epidemic simulations, preventive interventions substantially modify the magnitude and timing of epidemic peaks, while in the wildfire simulations relatively small variations in vegetation connectivity, wind conditions, and burning duration can lead to pronounced differences in fire extent and propagation patterns.

The comparison further demonstrates the flexibility of agent-based modeling for investigating complex dynamic systems characterized by heterogeneity, local interactions, and nonlinear behavior. By allowing individual entities to operate autonomously and interact according to predefined rules, agent-based models provide a flexible framework for studying spreading processes that are often difficult to analyze analytically or reproduce experimentally.

TABLE V. QUALITATIVE EFFECTS OF KEY PARAMETERS IN THE INVESTIGATED SIMULATION SCENARIOS

Model	Parameter	Observed Effect in the Investigated Scenario
Influenza	Mask Usage	Reduces transmission probability and delays epidemic peak
Influenza	Vaccination Coverage	Decreases the number of infectious individuals and increases population immunity
Influenza	Isolation Rate	Limits contacts after infection but acts with a delay
Wildfire	Forest Density	Determines connectivity among trees and extent of fire spread
Wildfire	Wind Speed	Accelerates directional fire propagation
Wildfire	Burning Duration	Influences secondary ignitions and final burned area

Table V summarizes the qualitative effects observed for the key parameters under the investigated simulation settings. In the epidemic model, the investigated mask, vaccination, and isolation scenarios produced different propagation outcomes under their respective parameter settings. Similarly, in the wildfire model, different combinations of forest density, wind conditions, and burning duration produced distinct fire propagation patterns. These observations are scenario-specific and should not be interpreted as a quantitative ranking of parameter influence. Nevertheless, they illustrate how agent-based simulation can support the comparative exploration of alternative parameter configurations in complex adaptive systems.

Several limitations should be acknowledged. The proposed models rely on simplified interaction mechanisms and do not incorporate empirical calibration, realistic mobility patterns, heterogeneous landscapes, or high-resolution meteorological datasets. Accordingly, the reported results should be interpreted as simulation-based observations under the investigated parameter settings rather than as empirically validated predictions of real-world epidemic or wildfire behavior. In addition, the present analysis is based on representative simulation runs and does not quantify stochastic variability through repeated runs or confidence intervals. Future work should employ multiple replications for each scenario and report statistical measures of variability to assess the robustness of the observed propagation patterns. The study does not include quantitative benchmarking against conventional domain-specific approaches, such as compartmental epidemic models or cellular automata-based wildfire models; therefore, no claims are made regarding superior predictive accuracy, computational efficiency, or model fidelity. Furthermore, the study focuses on representative simulation scenarios rather than predictive forecasting. Further extensions may integrate real-world datasets and calibration procedures, employ systematic parameter sweeps, sensitivity analyses, and factorial experimental designs to quantify the individual and interaction effects of forest density, wind conditions, and burning duration, and extend the methodology to additional categories of spreading phenomena, including information diffusion and cascading infrastructure failures.

VI. CONCLUSION

This study presented a common agent-based modeling methodology, implemented in the AnyLogic simulation environment, for investigating different spreading phenomena in complex adaptive systems. The proposed methodology was demonstrated through two representative application domains: influenza transmission within a higher education institution and wildfire propagation in a forest environment. Although the underlying mechanisms of these processes differ substantially, both models were developed using consistent modeling principles based on autonomous agents, statechart-driven behavior, stochastic state transitions, local interactions, and scenario-based experimentation.

The influenza simulations demonstrated that the investigated mask, vaccination, and isolation scenarios reduced epidemic intensity compared with the baseline under the specific parameter settings considered. Because the interventions were implemented using different mechanisms and coverage levels, these results represent scenario-specific outcomes rather than a normalized comparison of intervention effectiveness. These findings illustrate the potential role of preventive measures in modifying epidemic dynamics within the simulated educational environment.

The wildfire simulations produced substantially different propagation patterns across the investigated combinations of vegetation density, wind conditions, and burning duration. Dense and sparse forest configurations showed markedly different fire-spread patterns, while the scenario including strong eastward wind exhibited directional propagation. Because multiple parameters varied across some scenarios, the observed differences should be interpreted as scenario-specific outcomes rather than as isolated effects of individual environmental factors.

The comparative analysis demonstrated that, despite representing different scientific domains, both models exhibit

common propagation characteristics arising from local interactions among autonomous agents operating under stochastic conditions. The successful application of the same modeling methodology to two substantially different spreading phenomena demonstrates its consistency and transferability across different application domains. This finding highlights the potential of agent-based modeling as a versatile simulation methodology for analyzing complex adaptive systems characterized by heterogeneous entities, nonlinear dynamics, and emergent behavior.

Future research may extend the proposed methodology by incorporating empirical calibration, realistic mobility patterns, heterogeneous landscapes, high-resolution environmental and meteorological datasets, and additional categories of spreading phenomena, such as information diffusion, cascading infrastructure failures, and ecological invasion processes. Such extensions would further improve the generality and practical applicability of the proposed modeling approach.

REFERENCES

[1] M. J. North and C. M. Macal, Managing Business Complexity: Discovering Strategic Solutions With Agent-Based Modeling and Simulation. New York, NY, USA: Oxford University Press, 2007.

[2] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation," J. Simul., vol. 4, no. 3, pp. 151–162, 2010.

[3] J. M. Epstein, Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton, NJ, USA: Princeton University Press, 2012.

[4] A. Borshchev, The Big Book of Simulation Modeling: Multimethod Modeling With AnyLogic 6. Chicago, IL, USA: AnyLogic North America, 2013.

[5] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," Proc. R. Soc. Lond. A, vol. 115, no. 772, pp. 700–721, 1927.

[6] H. W. Hethcote, "The mathematics of infectious diseases," SIAM Rev., vol. 42, no. 4, pp. 599–653, 2000.

[7] L. Danon et al., "Networks and the epidemiology of infectious disease," Interdiscip. Perspect. Infect. Dis., vol. 2011, Art. no. 284909, 2011.

[8] R. M. Anderson and R. M. May, Infectious Diseases of Humans: Dynamics and Control. Oxford, U.K.: Oxford University Press, 1991.

[9] N. M. Ferguson et al., "Strategies for mitigating an influenza pandemic," Nature, vol. 442, no. 7101, pp. 448–452, 2006.

[10] I. M. Longini, Jr., et al., "Containing pandemic influenza at the source," Science, vol. 309, no. 5737, pp. 1083–1087, 2005.

[11] M. E. Halloran et al., "Modeling targeted layered containment of an influenza pandemic in the United States," Proc. Natl. Acad. Sci. USA, vol. 105, no. 12, pp. 4639–4644, 2008.

[12] P. Fine, K. Eames, and D. L. Heymann, "Herd immunity: A rough guide," Clin. Infect. Dis., vol. 52, no. 7, pp. 911–916, 2011.

[13] S. Cauchemez et al., "Closure of schools during an influenza pandemic," Lancet Infect. Dis., vol. 9, no. 8, pp. 473–481, 2009.

[14] J. T. Wu et al., "School closure and mitigation of pandemic (H1N1) 2009, Hong Kong," Emerg. Infect. Dis., vol. 16, no. 3, pp. 538–541, 2010.

[15] C. Fraser, S. Riley, R. M. Anderson, and N. M. Ferguson, "Factors that make an infectious disease outbreak controllable," Proc. Natl. Acad. Sci. USA, vol. 101, no. 16, pp. 6146–6151, 2004.

[16] N. Hoertel et al., "A stochastic agent-based model of the SARS-CoV-2 epidemic in France," Nat. Med., vol. 26, no. 9, pp. 1417–1421, 2020.

[17] R. Hinch et al., "OpenABM-Covid19—An agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing," PLoS Comput. Biol., vol. 17, no. 7, Art. no. e1009146, 2021.

[18] X. Zhang et al., "Agent-based modeling of epidemics: Approaches, applications, and future directions," Technologies, vol. 13, no. 7, Art. no. 272, 2025.

[19] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," Rev. Mod. Phys., vol. 87, no. 3, pp. 925–979, 2015.

[20] M. Pósfai and A.-L. Barabási, Network Science. Cambridge, U.K.: Cambridge University Press, 2016.

[21] B. Drossel and F. Schwabl, "Self-organized critical forest-fire model," Phys. Rev. Lett., vol. 69, no. 11, pp. 1629–1632, 1992.

[22] B. D. Malamud, G. Morein, and D. L. Turcotte, "Forest fires: An example of self-organized critical behavior," Science, vol. 281, no. 5384, pp. 1840–1842, 1998.

[23] D. Stauffer and A. Aharony, Introduction to Percolation Theory, 2nd ed. London, U.K.: Taylor & Francis, 1994.

[24] U. Wilensky, NetLogo Fire Model. Evanston, IL, USA: Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999.

[25] I. Karafyllidis and A. Thanailakis, "A model for predicting forest fire spreading using cellular automata," Ecol. Model., vol. 99, no. 1, pp. 87–97, 1997.

[26] C. Hackett, R. de Andrade Moral, G. Misra, T. McCarthy, and C. Markham, "An efficient method to simulate wildfire propagation using irregular grids," Nat. Hazards Earth Syst. Sci., vol. 25, pp. 2909–2928, 2025.

[27] K. Kuhnen et al., "Simulating wildfire propagation in Central Europe: Evaluation of input data variation and parameter adjustment for the PhyFire model," Ecol. Model., vol. 518, Art. no. 111615, 2026.

[28] E. Cardoso, D. X. Viegas, and A. G. Lopes, "A systematic review of models for fire spread in wildfires by spotting," Fire, vol. 8, no. 10, Art. no. 392, 2025.

[29] M. Ljubenovska, L. Koceva Lazarova, N. Stojkovikj, A. Stojanova, and M. Miteva, "Mathematical modeling of COVID-19 virus," in Proc. CIIT 2021, 2021.

[30] N. Stojkovikj et al., "Application of agent-based modelling in learning process," Informatica, vol. 48, no. 1, 2024.

[31] B. Zlatanovska et al., "Modeling of some chaotic systems with AnyLogic software," TEM J., vol. 7, no. 2, pp. 465–470, 2018.

[32] N. Stojkovikj, M. Kukusheva Paneva, B. Zlatanovska, and L. Koceva Lazarova, "Modelling, analysis and simulation of tuberculosis," Asian-Eur. J. Math., vol. 17, no. 9, Art. no. 2450073, 2024, doi: 10.1142/S1793557124500736.

[33] The AnyLogic Company, "AnyLogic simulation software." [Online]. Available: https://www.anylogic.com/. Accessed: Jun. 25, 2026.

Play-Centric Design and Usability Evaluation of a Block-Based Programming Serious Game for Low Vision Children

Huang Sheng, Chau Kien Tsong*, Wan Ahmad Jaafar Wan Yahaya
Universiti Sains Malaysia, Malaysia

***Abstract*—Serious games are effective tools for learning beyond entertainment, yet children with low vision face obstacles when playing block-based programming games because these environments rely on small graphical elements, low colour contrast, and drag-and-drop interaction, a barrier specific to block-based programming that does not arise in other game genres. This study presents the design and usability evaluation of a block-based programming serious game for teaching sequences, loops, and conditionals to low-vision children aged seven years old. Accessibility elements were implemented in the prototype, namely screen design, multimedia, keyboard-based interaction replacing drag-and-drop, and language. Usability testing and a pre-test and post-test were conducted with sixteen children in matched experimental and control groups of eight, the control group using a feature-reduced baseline version, with results reported descriptively. The accessible version obtained a Good overall usability score of 3.80, against a Weak score of 1.65 for the baseline. Only 25 per cent of control participants independently reached all three target concepts, compared with 100 per cent in the accessible version. Because only baseline participants who reached the target concepts were compared, the learning gain difference is a conservative estimate of the accessibility effect.**

Keywords*—Serious game; block-based programming; low vision; accessibility; usability testing; computational thinking; assistive technology

I. Introduction

Nowadays, information and communication technologies (ICT) are used in education. We can find a lot of educational applications that can help young learners learn. Several ICT applications are on the market, such as e-books, multimedia courseware, and games [1][2]. Gaming is a very popular activity enjoyed by children. Thus, a serious game is the most promising tool for helping children learn. A serious game is used beyond entertainment [1]. Serious games are powerful tools for building language, science, math, and computing knowledge and skills. This educational value has been demonstrated across contexts including early childhood education [22] and STEM learning more broadly [24]. Despite a growing number of studies on using serious games for learning, children with visual impairments have obstacles when playing the games. There are two categories of visual impairment: blindness and low vision. Low vision children have a visual impairments that can be assisted with assistive technology. The World Health Organisation defines low vision as visual acuity between 20/70 and 20/400, with a possible correction of 20 degrees or less [31]. Low vision children have obstacles accessing graphical elements in games. Prior studies indicate that low vision children face several accessibility issues, such as visual, animation, audio, and navigation obstacles [9], [11]. This is consistent with broader evidence that low vision affects reading behaviour [17] and the interpretation of visual information more generally [18]. They need to look at graphical elements from a very close distance because the size of the text and graphics is small. The colour contrast between the graphics and background is often low. They also have difficulty navigating the menus and buttons in games, and fast animation movements affect their vision when playing games.

Among educational applications, block-based programming environments such as Scratch have become the most popular tool for introducing young children to computational thinking, because they allow children to build instructions by assembling visual blocks instead of typing code [4], [21]. The use of such tools to build computational thinking and programming skills has been demonstrated across K-12 educational settings [25], [26]. Unlike most other educational applications, however, block-based programming environments depend on an interaction method that other serious game genres do not require: drag-and-drop, where children must click, hold, and drag a block precisely into place to construct a program. This dependence on drag-and-drop is a barrier specific to block-based programming. This is consistent with accessibility issues documented across block-based programming environments more broadly [20]. A mathematics or language serious game can be made usable for low vision children through screen design and multimedia adaptations alone [3], but a block-based programming game remains unusable if children cannot physically perform the drag-and-drop action, regardless of how large or high-contrast the blocks are made. Thus, to ensure low-vision children can play the game, a block-based programming serious game for learning sequences, loops, and conditionals is developed for low-vision children aged seven years old. The game consists of three levels of programming tasks, based on introductory computational thinking content appropriate for Year 1 pupils, namely ordering instructions (sequences), repeating instructions (loops), and if-then decision making (conditionals). These constructs are consistent with the core computational-thinking skills assessed in prior block-based programming research [14].

Serious games should be easily accessible to low vision children. The game design should be flexible, with an interface adapted to the accessibility requirements of low vision children.

*Corresponding author.

To enhance the game's accessibility for low vision children, accessibility elements have been implemented in the serious game prototype. Those elements are screen design (buttons, menus, and navigation), multimedia (text, graphics, and audio), interaction method, and language. The children can set the appropriate background colour and contrast level based on their vision level. The text size is large and adjustable. The background colour has a high colour contrast, and bright colours and large graphics are implemented in the game. Background audio and navigation cues are provided so children can identify their position on the screen and understand the game's instructions without relying on vision alone. Most importantly, the conventional drag-and-drop interaction used to assemble programming blocks is replaced with a keyboard-based selection method: children navigate the interface using arrow keys, and press Enter to select and place a block, removing the need for precise cursor control. This design choice is consistent with accessible input approaches explored for other populations with fine motor or visual constraints [19]. A feature-reduced baseline version of the same game, without these accessibility elements, is also developed to serve as a comparison during usability testing.

Upon completion of the serious game, usability testing is conducted to identify the accessibility of the serious game for low-vision children based on the usability level, and a pre-test/post-test is conducted to identify whether the game helps low-vision children understand the target programming concepts.

Thus, this study presents the design and usability testing of a block-based programming serious game prototype developed specifically for low vision children aged seven years old. The serious game is developed based on the accessibility requirements of low vision children, refined through pilot testing and expert heuristic evaluation prior to user testing. The effectiveness, efficiency, and user satisfaction of the serious game are tested in the usability testing, and the learning outcome of the game is tested using a pre-test and post-test benchmarked against a feature-reduced baseline version. The observation technique is used for analysing the serious game. The usability testing aims to identify the serious game's usability level and learning effectiveness for low-vision children.

This study is organised as follows: Section I presents the introduction, while Section II describes the related work. Section III describes the methodology, and Section IV and Section V discuss the results and discussion, respectively, while Section VI presents the conclusion and plan for future work.

II. Related Work

The International Organization for Standardization (ISO) 9241-11 (1998) defines usability as the ability of a product to be used by users to accomplish certain goals effectively, efficiently, and with user satisfaction [30]. Usability also refers to whether a product is easy to use. Based on the ISO 9241-11 definition, usability consists of the following components:

- Effectiveness: The accuracy and completeness with which a user can achieve a certain goal in a certain environment.
- Efficiency: The effort the user gives to complete a task and achieve the objective of using an application or system.
- User Satisfaction: User comfort and acceptance of the system.

Usability testing is conducted to identify usability problems before releasing the serious game in its real context. There is some overlap between the concepts of usability and accessibility: accessibility enables users to access ICT applications independently without accounting for their disability, while usability refers to the ability of the ICT application to carry out the intended function effectively, efficiently, and with satisfaction when used by users in general [30]. Therefore, accessibility specifically refers to disabled users, while usability refers to the general population. Usability testing is also considered suitable for assessing ICT applications for accessibility [7].

Prior research has proposed a serious-game accessibility design model consisting of three components, namely accessibility, game design, and pedagogy [11], and implemented accessibility elements including screen design, multimedia, object motion, and language in a mathematics serious game for low vision children, refined using a play-centric design approach through a low-fidelity prototype and pilot feedback from low vision children [8]. Usability testing of this game, conducted through direct observation with fifteen low vision children, reported good usability scores across effectiveness (4.74), efficiency (4.55), and user satisfaction (4.92), with an overall usability score of 4.76 [3]. These findings show that accessibility elements such as screen design, multimedia, language, and object motion are important in increasing the usability of serious games for low vision children. However, this study focused on a mathematics game, in which children select and tap answers on screen; it does not address the additional interaction barrier present in block-based programming, where children are required to drag and connect blocks to construct a sequence of instructions, nor does it assess whether use of the game was associated with a measurable gain in the target academic content.

Block-based programming languages have become the dominant approach to introductory programming education because they lower the syntactic barrier to entry for young learners [4]. Prior efforts to make block-based programming accessible have largely targeted children who are blind: Blocks4All, an iPad-based block-based programming environment, replaces the visual programming canvas with touchscreen exploration and VoiceOver audio feedback, and was evaluated with children who are blind or have visual impairments controlling a physical robot [5]. This and related approaches are designed around the assumption that the user has no usable vision, and therefore do not directly address the residual-vision use case of low vision children, for whom enlarged and high-contrast visual elements, combined with reduced dependence on fine motor precision rather than a fully non-visual interface, are more appropriate.

A representative sample of users must conduct usability testing because users are different and have different problems [6]. Suitable usability testing methods should be considered when planning testing with children. Usability testing with children consists of introspection, direct observation, thinking aloud, and interaction. Direct observation is used to observe users as they perform their tasks, and this technique has been recognised as one of the best techniques for usability testing with young children, since it does not depend on children's ability to verbalise their experience [6], [15]. Satisfaction with children is commonly measured using simplified emoticon-based scales, such as like, neutral, and dislike, to observe the user's reaction to the game [3].

Accessibility research more broadly has also been reported within this journal, for example in evaluations of web accessibility barriers [9] and usability evaluations of interfaces for specific user populations [12][13], reflecting the value of usability testing as a method for surfacing accessibility problems that are not apparent from design review alone. While prior studies have evaluated the usability of serious games for low vision children, usability testing alone does not establish whether the target educational content was learned. Coupling usability testing with a pre-test/post-test learning outcome assessment allows a study to determine not only whether low vision children can use a serious game, but whether using it is associated with a gain in the knowledge the game is intended to teach.

III. METHODOLOGY

The usability testing method used in this study is direct observation. The observation method is conducted to evaluate the user experience of the product by observing users carrying out the tasks provided. Observations are conducted based on effectiveness, efficiency, and user satisfaction. User satisfaction is subjective; thus, interviews are also conducted with low-vision children to obtain feedback regarding their satisfaction level with the tested serious game. During the interview, emoticons (like, neutral, dislike) are prepared to observe the user's reaction to the game. In addition to usability testing, a pre-test and post-test are conducted to identify whether the game is associated with a gain in understanding of sequences, loops, and conditionals. The overall study follows a three-phase process: 1) design and iterative refinement of the game prototype, incorporating pilot testing with children and expert heuristic evaluation; 2) recruitment and assignment of participants; and 3) usability and learning-outcome testing.

A. Game Design and Prototype Refinement

The game is designed following a three-component accessibility model, namely accessibility, game design, and pedagogy [11]. A play-centric design approach is adopted, beginning with a low-fidelity prototype refined through pilot feedback from three to five low-vision children, consistent with the play-centric prototyping approach used in prior serious-game design research for this population [8].

Table I shows the accessibility elements implemented in the accessible version of the game, and the corresponding elements omitted in the feature-reduced baseline version used for comparison during testing.

Both versions shared the same level content, learning objectives, and progression structure. They differed only in the presence or absence of the full set of accessibility elements summarised in Table I (screen design, multimedia, interaction method, and text). Consequently, any systematic differences observed between the experimental and control groups can be attributed to the accessibility package, rather than to the keyboard-based interaction method in isolation.

TABLE I. ACCESSIBILITY ELEMENTS IN THE ACCESSIBLE AND BASELINE VERSIONS

Element	Accessible version	Baseline version
Screen design	Large, adjustable buttons and menus	Standard, fixed-size buttons and menus
Multimedia	High-contrast graphics, adjustable colour scheme, audio feedback	Standard-contrast graphics, no audio feedback
Interaction	Keyboard-based selection (arrow keys + Enter)	Mouse-based drag-and-drop
Text	Large, adjustable font size	Standard, fixed font size

Note: The accessible version differs from the baseline version by the full set of accessibility adaptations listed above. Differences in outcomes between the two versions are therefore attributable to the combined accessibility package rather than to any single element in isolation.

The baseline version is intended as a credible feature-reduced comparator rather than a deliberately inaccessible interface: it retains the same learning content, level structure, narrative theme, and progression sequence as the accessible version, and systematically omits only the accessibility elements listed in Table I (large, adjustable buttons and menus; high-contrast graphics; adjustable colour schemes; audio feedback; keyboard-based selection; and large, adjustable text). This design allows any observed differences in usability scores, concept-exposure rates, and learning gains between the two groups to be attributed to the presence or absence of the accessibility package, rather than to differences in pedagogical content or task difficulty.

Following pilot testing, the refined prototype is submitted to an expert heuristic evaluation prior to full user testing. Four experts in human-computer interaction, special education, and game design independently evaluate the prototype using two heuristic sets: game heuristics (challenge, reward, goals, storyline) and pedagogical heuristics (learning objectives, learning theory, content alignment), and complete a structured evaluation questionnaire with improvement recommendations. This heuristic evaluation approach follows methods developed specifically for assessing usability and user experience in children's digital tools [16]. Recommendations arising from this evaluation are consolidated and used to guide a second round of prototype refinement before the prototype is used in the user testing phase.

The serious game user interface, incorporating the accessibility elements described above, is depicted in Fig. 1.

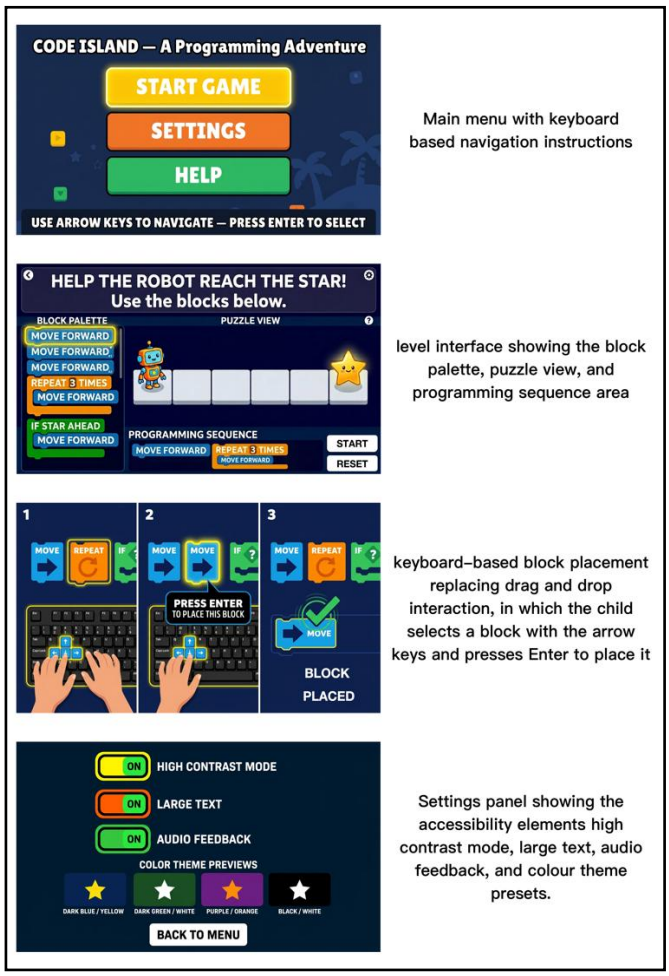


Fig. 1. Serious game user interface.

B. Participants

This study uses a selected study sample recruited from a special education school. Low vision is defined according to World Health Organisation criteria as visual acuity between 20/70 and 20/400, with a possible correction of 20 degrees or less [31], verified through participants' school medical records. This testing involves low vision children aged seven years old, assigned to an experimental group using the accessible version of the game and a control group using the feature-reduced baseline version, matched by age, prior programming or game experience, and level of visual acuity. This study was conducted in accordance with the principles of the Declaration of Helsinki. Ethical approval for this multi-centre study was obtained from the institutional review boards corresponding to the authors' respective affiliations: the Jawatankuasa Etika Penyelidikan Manusia, Universiti Sains Malaysia (No. USM/JEPeM/PP/26070645), for all authors from Universiti Sains Malaysia, and the South China Normal University Ethics Approval Certificate for Research Involving Human Subjects (No. SCNU-2026-039), for the author also affiliated with South China Normal University. To conduct this testing, ethics that the researcher must follow are observed: consent from parents or guardians is required before testing to ensure the study meets research guidelines and ethics, the purpose of the research is explained to the teachers involved, and children are informed of their right to withdraw at any time. Both groups receive the same fixed session duration and the same level of researcher presence throughout testing, so that any difference observed between groups can be attributed to the accessibility adaptations rather than to differences in attention received or to the basic novelty of encountering an unfamiliar digital tool.

C. Usability Testing Procedures

The activities involved in usability testing include determining the testing objective, preparing the instruments and testing tools, identifying the task scenario, preparing the testing checklist, and measuring usability. The testing procedure is further explained in the following sections.

1) Determination of testing objective: This usability testing is conducted to verify the accessibility of the serious game to low-vision children. The evaluation is conducted based on the ISO 9241-11 definition, which covers effectiveness, efficiency, and user satisfaction concerning the serious game [30].

2) Testing instruments and testing tools preparation: The testing instruments consist of the accessible and baseline serious game prototypes, a usability testing task list, an observation checklist, and audio and screen recording tools.

3) Identify task scenarios: Usability testing is conducted by giving the user a task scenario while the user plays the game, prepared according to the game's screens, such as the start screen, the level screens, and the task screen within each level.

4) Testing checklist preparation: A checklist is prepared based on the usability constructs of effectiveness, efficiency, and user satisfaction [30], covering the accessibility elements shown in Table I.

5) Measurement of usability testing: The measurement of effectiveness, efficiency, and user satisfaction is conducted using a usability score, following the scoring approach used in prior serious-game usability research [3].

6) Effectiveness measurement: If a task is completed successfully, a score of 5.0 is marked on the effectiveness checklist item. If the task is half completed, a score of 2.5 is marked; if the task is not completed, a score of 0 is marked. The usability score for effectiveness, X, is formulated as:

$$X = \frac{\Sigma\ score\ for\ each\ effectiveness\ checklist\ item}{total\ effectiveness\ checklist\ items} \quad (1)$$

7) Efficiency measurement: Efficiency is measured based on the time taken to complete the task relative to the average task completion time. The average task completion time used as the scoring benchmark is established separately for each version (accessible and baseline) from the pilot study described in Section III-C6, to account for the difference in interaction speed between the keyboard-based and mouse-based interaction methods; each level's benchmark is computed as the mean completion time recorded for that level across the pilot sample. If the time taken is equal to or faster than the average, a score of 5.0 is marked; if slower, a score of 2.5 is marked; if the task fails, a score of 0 is marked. The usability score for efficiency, X, is formulated as:

$$X = \frac{\sum \ score \ for \ each \ efficiency \ checklist \ item}{total \ efficiency \ checklist \ items} \quad (2)$$

8) User satisfaction measurement: A score of 5.0 is marked if a child's reaction is "like"; 2.5 if "neutral"; and 0 if "dislike". The usability score for user satisfaction, X, is formulated as:

$$X = \frac{\sum \ score \ for \ each \ satisfaction \ checklist \ item}{total \ satisfaction \ checklist \ items} \quad (3)$$

The average value of each usability category is considered the score for that category, and the overall usability score is the average of the three category scores. This scoring approach is consistent with psychometric usability and acceptability scales developed for children and adolescent users [23]. Table II shows the interpretation of the overall usability score, following the interpretation bands used in prior research [3].

TABLE II. USABILITY SCORE INTERPRETATION

Weak	Moderately Good	Good
$0 \leq x \leq 2.49$	$2.5 \leq x \leq 3.49$	$3.5 \leq x \leq 5.0$

9) Implementation of usability test: A pilot study is conducted on the instrument to test its reliability before it is used in the actual testing, involving three to five low-vision children, distinct from the pilot testing conducted during prototype refinement. During usability testing, children play the game with minimal researcher presence; no instructional assistance is provided during the session, so that the effectiveness, efficiency, and satisfaction scores reflect the accessibility of the game itself rather than researcher intervention. Because young children may not spontaneously ask to stop an activity even when struggling, the session is not left to run purely on the child's initiative to withdraw: the researcher proactively pauses the session and checks in with the child if either of the following is observed: no measurable progress for three consecutive minutes, or five consecutive failed attempts at the same step without progression, at which point the child is asked whether they would like to continue or take a break, and a break or early termination is offered accordingly. This check-in is in addition to, not instead of, immediate termination upon overt signs of distress. Control group participants are offered the opportunity to try the accessible version of the game after their session has ended. A screen recorder and camera are used to record testing sessions, and researchers observe and record children's actions on the provided task scenario checklist, which is then transferred to the usability checklist. The data collected are analysed using a descriptive analysis method.

D. Learning Outcome Assessment

In addition to usability testing, a pre-test and post-test are administered to identify whether use of the game is associated with a gain in understanding of sequences, loops, and conditionals.

1) Instrument: Each pre-test and post-test consisted of nine items organised into three equal sections corresponding to the three target concepts: three items on sequences, in which children arranged or identified the correct order of a short series of action steps to complete a simple path; three items on loops, in which children identified or constructed the correct repeating pattern needed to complete a repeated action; and three items on conditionals, in which children selected the appropriate if-then rule to make a character respond differently under two conditions. Each correct response was awarded one point, with no partial credit for individual items, yielding a maximum possible score of 9 on both the pre-test and the post-test. The pre-test and post-test are administered one-on-one, in oral format, accompanied by large, high-contrast visual aids, consistent with the accessibility elements applied to the game itself. Equivalent item sets are used for the pre-test and post-test, differing only in surface-level details such as character names, object colours, and path layouts, to reduce practice effects. The item set is reviewed by a special education teacher and a computer science educator for alignment with the target concepts, age-appropriateness, and clarity for oral administration.

2) Procedure: Both the experimental group and the control group complete the pre-test before their assigned game session and the post-test immediately afterwards. Learning gain is calculated as the difference between the post-test and pre-test scores.

3) Concept-exposure criterion and analysis: For the experimental group, all participants are expected to reach the levels covering all three core concepts within the session and are included in the learning gain analysis. For the control group, only participants who independently reach these levels are included; participants who do not reach concept exposure are excluded from the learning gain comparison, but their outcome-how many control participants did and did not reach concept exposure, and at which level they stopped is retained and reported as usability evidence in its own right, since it directly quantifies the severity of the barriers present in the baseline environment. If the number of control group participants who independently reach concept exposure is fewer than three, group-level means are not reported for this subgroup; instead, the individual pre-test and post-test scores of these participants are reported as case data. Because the participants who reach concept exposure independently are not representative of the control group as assigned, they are more likely to be those with relatively milder visual impairment, better fine motor control, or prior exposure to similar interfaces; any comparison between this subgroup and the experimental group is interpreted as a conservative estimate of the accessibility effect: the true gap between the accessible and baseline conditions is likely larger than the reported comparison suggests. Given the exploratory nature of this study and the anticipated small sample size, learning gain and usability scores are reported descriptively (means, standard deviations, frequencies); no inferential statistical tests are performed. A non-parametric procedure such as the Wilcoxon signed-rank test was considered as an exploratory measure for the matched-pair learning-gain comparison; however, it is not feasible here because six of the

eight matched pairs are incomplete for this comparison, as the corresponding control-group participant did not reach concept exposure (Section IV-D). A matched-pairs analysis restricted to the two complete pairs would be too small to yield an interpretable test statistic, even as an exploratory measure, and is therefore not reported.

IV. Results

This section reports the results of the usability testing and the pre-test/post-test learning outcome assessment. Consistent with the exploratory design and small sample size described in Section III-D3, all usability scores, learning gains, and event counts are reported descriptively using means, standard deviations, and frequencies; no inferential statistical tests are performed. The results are presented in the order of the participant characteristics, the expert heuristic evaluation conducted prior to user testing, the usability scores, the concept-exposure outcomes for the control group, the learning outcome results, the termination-criterion events, and the inter-rater reliability of the learning-outcome scoring.

A. Participants

Table III shows the demographic and matching characteristics of the participants recruited for this study.

TABLE III. Participant Characteristics and Group Assignment

ID	Group	Matched pair	Age (months)	Visual acuity (decimal)	Prior experience	Sex
P01	Experimental	M01	83	0.10	Minimal	M
P02	Control	M01	84	0.12	Minimal	M
P03	Experimental	M02	87	0.08	None	F
P04	Control	M02	86	0.10	None	F
P05	Experimental	M03	80	0.15	Some	M
P06	Control	M03	81	0.13	Some	M
P07	Experimental	M04	89	0.05	Minimal	F
P08	Control	M04	88	0.07	Minimal	F
P09	Experimental	M05	84	0.11	None	M
P10	Control	M05	83	0.09	None	M
P11	Experimental	M06	86	0.14	Minimal	F
P12	Control	M06	87	0.12	Minimal	F
P13	Experimental	M07	79	0.10	Some	M
P14	Control	M07	80	0.11	Some	M
P15	Experimental	M08	90	0.06	None	F
P16	Control	M08	89	0.08	None	F
Total (Experimental, n=8)		8 pairs	Range 79–90 (M≈84.8)	0.05–0.15, all WHO-confirmed low vision	None=3, Minimal=3, Some=2, (see note)	M=4, F=4
Total (Control, n=8)			Range 80–89 (M≈84.8)	0.07–0.13, all WHO-confirmed low vision	None=3, Minimal=3, Some=2, (see note)	M=4, F=4

Note: All 16 participants were assigned in eight age-, sex-, prior-experience-, and visual-acuity-matched pairs (one experimental, one control participant per pair), consistent with the matching procedure described in Section III-B. Prior experience distribution across the full sample (both groups combined) was None=6, Minimal=6, Some=4. Visual acuity was confirmed against school medical records for all participants and fell within the WHO low vision range in every case.

B. Expert Heuristic Evaluation

Table IV shows the results of the expert heuristic evaluation conducted prior to user testing, covering game heuristics and pedagogical heuristics, together with the resulting prototype refinements.

As shown in Table IV, all seven heuristic dimensions received mean expert ratings between 3.8 and 4.5 on the 5-point scale, indicating that the prototype was rated favourably across both the game and pedagogical heuristics prior to user testing. The lowest-rated dimension was Challenge (3.8), reflecting expert concern about the difficulty increase between the Loops and Conditionals concepts, which was addressed by adding an intermediate practice level before the Conditionals concept. The Goals and Learning objectives dimensions received the highest ratings (4.5 each), and the associated recommendations, together with those for the remaining dimensions, were implemented as concrete refinements to the prototype before the user testing phase. Because each recommendation was translated into a specific design change, the version subsequently evaluated in the usability testing already incorporated the improvements identified during the heuristic evaluation.

TABLE IV. EXPERT HEURISTIC EVALUATION RESULTS

Heuristic dimension	Mean expert rating	Key recommendation	Refinement implemented
Challenge	3.8 (0.5)	Reduce the difficulty jump from loops to conditionals	Added an intermediate practice level before conditionals
Reward	4.3 (0.5)	Make success feedback more salient for low vision users	Added a distinct success chime and large green checkmark animation
Goals	4.5 (0.6)	Clarify the learning goal of each level more explicitly	Added a short audio and large-text goal statement at the start of every level
Storyline	4.0 (0.8)	Strengthen narrative continuity across the three concepts	Linked levels with a simple character-progress story and a consistent visual theme
Learning objectives	4.5 (0.6)	Make the three target concepts more explicitly visible	Display the current concept name (Sequences, Loops, Conditionals) in large high-contrast text on each level screen
Learning theory alignment	4.2 (0.5)	Better support progressive scaffolding of computational thinking constructs.	Inserted guided practice tasks that reuse previous concept blocks before introducing the next
Content alignment	4.3 (0.5)	Ensure block vocabulary matches Year-1 curriculum language	Simplified block labels and added optional audio read-aloud of each block name

Note: Each dimension was rated by the four experts on a 5-point scale (1 = poor, 5 = excellent); values are reported as mean (SD) across experts. Game heuristics and pedagogical heuristics were evaluated using two separate structured checklists, as described in Section III-A. Each key recommendation was addressed through the corresponding refinement prior to the user testing phase.

C. Usability Results

Table V shows the effectiveness, efficiency, and user satisfaction scores for the experimental and control groups, calculated using Eq. (1)-(3) and interpreted using the bands in Table II, reported per level.

TABLE V. EFFECTIVENESS, EFFICIENCY, AND SATISFACTION SCORES BY GROUP AND LEVEL

Group	Level	Effectiveness (SD)	Efficiency (SD)	Satisfaction (SD)	Overall score	Interpretation
Experimental	1 – Sequences	4.54 (0.81)	4.22 (0.99)	4.54 (0.81)	4.43	Good
Experimental	2 – Loops	3.90 (1.11)	3.59 (1.19)	3.90 (1.11)	3.80	Good
Experimental	3 – Conditionals	3.29 (1.23)	2.97 (1.27)	3.29 (1.23)	3.18	Moderately Good
Experimental	Overall	3.91 (1.12)	3.60 (1.18)	3.91 (1.12)	3.80	Good
Control	1 – Sequences	3.12 (1.25)	2.50 (1.31)	2.81 (1.28)	2.81	Moderately Good
Control	2 – Loops	1.88 (1.25)	1.25 (1.16)	1.56 (1.21)	1.56	Weak
Control	3 – Conditionals	0.78 (0.99)	0.47 (0.81)	0.47 (0.81)	0.57	Weak
Control	Overall	1.93 (1.34)	1.41 (1.24)	1.62 (1.29)	1.65	Weak

Note: Efficiency scores are calculated using effective gameplay time, not total session duration. Sub-level scores were recorded for two task instances per concept; the values above are the mean of the two sub-level scores per concept per group. Control-group scores are computed across the full control group (n=8), following the effectiveness/efficiency/satisfaction scoring rule in Section III-C5 in which a non-completed task is scored 0; these scores therefore reflect both the performance of participants who progressed through a level and the zero scores of participants who did not reach it, and are the appropriate usability evidence for the group as assigned. The n=2 control-independent subgroup (participants who reached all three concepts unaided) is reported separately in Section IV-E for the learning-outcome comparison and is not the basis for this table. In the control group, the number of participants completing each level was 6 at Level 1, 4 at Level 2, 3 at Level 3, and 2 at each of Levels 4, 5, and 6, consistent with the per-participant stopping levels reported in Table VI.

As shown in Table V, the experimental group obtained an overall usability score of 3.80, which falls within the good band, whereas the control group obtained an overall score of 1.65, which falls within the Weak band. The experimental group scored in the good band for the Sequences (4.43) and Loops (3.80) concepts, declining to the Moderately Good band for the Conditionals concept (3.18). The control group, in contrast, scored in the Moderately Good band only for Sequences (2.81) and fell into the Weak band for both Loops (1.56) and Conditionals (0.57). This pattern indicates that usability declined with increasing programming-concept complexity in both groups, but that the decline was substantially steeper for the control group, which used the feature-reduced baseline version without the accessibility elements listed in Table I.

The same downward trend is observed across all three usability components. For the experimental group, effectiveness declined from 4.54 at Sequences to 3.29 at Conditionals, efficiency from 4.22 to 2.97, and satisfaction from 4.54 to 3.29. For the control group, the corresponding declines were steeper, with effectiveness falling from 3.12 to 0.78, efficiency from 2.50 to 0.47, and satisfaction from 2.81 to 0.47. Across all three components and all three concepts, the experimental group scored higher than the control group, consistent with the accessibility elements contributing to usability for low-vision children playing a block-based programming game.

Effective gameplay time, excluding pauses and check-ins, averaged 24 min 15 s for the experimental group and 16 min 52 s for the control group, out of an equal total session duration of approximately 28 min 40 s for both groups. The two groups therefore received effectively equal session time, but the control group spent proportionally more of that time in pauses, check-ins, and early termination, with active gameplay accounting for 58.9% of the session for the control group compared with 84.5% for the experimental group. Because the efficiency scores in Table V are computed from effective gameplay time rather than total session duration, this difference is reflected in the efficiency component but does not arise from unequal time allocation between groups.

D. Concept Exposure in the Control Group

Table VI shows, for the control group, whether each participant independently reached the levels covering all three core concepts within the session; this outcome is reported as a usability finding, not merely as a precondition for inclusion in the learning-gain analysis.

As shown in Table VI, only two of the eight control-group participants (25%) independently reached the levels covering all three core programming concepts within the session. The remaining six participants stopped at an earlier level: two stopped at Level 1, two at Level 2, one at Level 3, and one at Level 4, with no participant stopping at Level 5. The attrition was concentrated within the Sequences concept itself, where four of the eight participants stopped at Level 1 or Level 2 without progressing to the Loops concept. This concept-exposure outcome is reported here as a usability finding, since the proportion of control participants unable to reach a given concept directly quantifies the severity of the barriers present in the unadopted baseline environment and does so in a way that the group-mean usability scores in Table V, which average completed and non-completed tasks together, do not fully convey.

TABLE VI. CONTROL GROUP CONCEPT-EXPOSURE OUTCOMES

ID	Reached all 3 concepts?	Last level reached	Reason for non-completion
P06	Yes	Level 6 (Conditionals)	—
P14	Yes	Level 6 (Conditionals)	—
P02	No	Level 1 (Sequences)	Unable to place the first block using mouse drag-and-drop after multiple attempts; stopped interacting..
P04	No	Level 1 (Sequences)	Could not distinguish the low-contrast blocks from the background; repeated wrong selections
P08	No	Level 2 (Sequences)	Lost track of the block sequence without audio cues; five consecutive failed attempts, session ended early after check-in
P10	No	Level 2 (Sequences)	Unable to complete drag-and-drop placement of the second block; disengaged after repeated failed attempts
P12	No	Level 3 (Loops)	Could not select and connect the loop block with the mouse; five consecutive failed attempts; session ended early after check-in
P16	No	Level 4 (Loops)	Lost track of the required loop repetitions without audio feedback; stopped progressing after repeated incorrect placements
Total reaching full exposure: 2 / 8 (25%)			

Note: The last level reached, and the reason for non-completion for each participant were recorded from the session observation checklists and screen recordings referenced in Section III-C6. Reasons are reported as observed interaction behaviours (for example, inability to complete a drag-and-drop placement, or repeated wrong selections of low-contrast blocks) rather than inferred internal states.

E. Learning Outcome Results

Table VII reports individual-level pre-test and post-test scores for all experimental group participants and for control group participants who independently reached concept exposure.

As shown in Table VII, all eight experimental-group participants recorded a positive learning gain. For the experimental group, the pre-test mean was 1.75 and the post-test mean was 7.44, giving a mean gain of 5.69 (SD=0.53), with individual gains ranging from +5.0 to +6.5. No experimental-group participant recorded a gain below +5.0, indicating a consistent improvement across the group.

Because the control-independent subgroup comprises only two participants, fewer than the pre-registered threshold of three specified in Section III-D3, the corresponding group-level learning-gain comparison is omitted, and only the individual case data are reported for this subgroup. Participants P06 and P14 recorded gains of +2.5 and +2.0, respectively. Both gains are smaller than that of any experimental-group participant, and both participants began from a pre-test range (1.5 to 2.0) comparable to that of several experimental-group participants, such as P01 and P11 (pre-test 2.5).

TABLE VII. INDIVIDUAL PRE-TEST/POST-TEST SCORES

ID	Group	Pre-test	Post-test	Gain
P01	Experimental	2.5	8.0	+5.5
P03	Experimental	1.0	7.5	+6.5
P05	Experimental	3.0	9.0	+6.0
P07	Experimental	2.0	7.0	+5.0
P09	Experimental	1.5	6.5	+5.0
P11	Experimental	2.5	8.5	+6.0
P13	Experimental	0.5	6.0	+5.5
P15	Experimental	1.0	7.0	+6.0
P06	Control-independent	2.0	4.5	+2.5
P14	Control-independent	1.5	3.5	+2.0

Note: The control-independent subgroup is not representative of the control group as originally assigned, since it comprises only the two participants (of eight) who were able to independently reach concept exposure in the baseline environment. Any comparison drawn between this subgroup and the experimental group is therefore interpreted as a conservative estimate of the accessibility effect (Section III-D3): the six control-group participants who did not reach concept exposure are absent from this comparison entirely, and their exclusion is itself evidence of a barrier severity that a group-mean comparison would not capture.

F. Termination-Criterion Events

Table VIII summarises how often the proactive termination criteria described in Section III-C6 were triggered during testing, and their outcomes.

TABLE VIII. TERMINATION-CRITERION EVENTS BY GROUP

Group	"3 min no progress" triggers	"5 consecutive failed attempts" triggers	Continued after check-in	Took a break	Session ended early
Experimental	3	2	4	1	0
Control	11	9	2	3	9

As shown in Table VIII, the control group triggered a proactive check-in substantially more often than the experimental group, with 20 triggers in total compared with 5. The outcome of these check-ins also differed between the two groups. In the experimental group, most check-ins led to the session continuing: of the five check-in occasions, four led to continuation and one led to a break, and no experimental-group session ended early. In the control group, by contrast, the session ended early on 9 of the 14 check-in occasions. All trigger events were logged with a timestamp during testing; for example, participant P04 triggered the "3-minute no-progress" criterion at 00:08:42 into the session. These timestamps are retained in the raw observation records but are not reported individually.

G. Inter-Rater Reliability

Before formal scoring, both the primary rater and the second rater underwent a short calibration session: the raters independently scored a small set of practice responses drawn from the pilot testing phase (not included in the main analysis) and then compared their scores, discussing discrepancies until consensus was reached on the application of the scoring rubric described in Section III-D (one point per item, nine items in total). After this calibration, to assess scoring reliability, 30% of all pre-test and post-test responses (stratified by participant and by test occasion) were independently re-scored by the second rater, who remained blinded to group assignment and to the primary rater's scores. Inter-rater agreement calculated with Cohen's κ was 0.82, indicating almost perfect agreement (Landis & Koch, 1977), supporting the reliability of the scoring approach used for the primary rater's results reported in Table VII.

V. Discussion

A. Usability of the Accessible Version

The experimental group achieved a Good overall usability rating (3.80) for the accessible version, consistent with the range reported for accessible serious games for low vision children in prior work [3]. However, the score was not uniform across concepts: usability was Good for Sequences (4.43) and Loops (3.80) but declined to Moderately Good for Conditionals (3.18), and the same downward pattern was present in effectiveness, efficiency, and satisfaction scores alike. This suggests that the accessibility elements applied large, high-contrast screen design, adjustable colour and text, and audio feedback supported usability effectively for lower-complexity tasks, but that usability nonetheless degrades as programming-concept complexity increases, even when accessibility barriers related to screen design and multimedia are addressed. This is consistent with prior findings that screen design elements such as menus and buttons contribute directly to interface usability [10]. Because the accessibility package (Table I) remained constant across all levels while only task complexity varied, the within-group decline is more plausibly attributable to the cognitive demands of conditional logic itself than to a breakdown in the accessibility adaptations at higher complexity. This is a favourable finding relative to the interaction barrier reported for drag-and-drop block assembly in prior block-based programming accessibility work [5]: the accessible version, which incorporates a keyboard-based selection method alongside its other accessibility adaptations, appears to have kept the interaction itself usable across all three concepts, with the residual difficulty concentrated in the conceptual content rather than the input method. Because the present design varies screen design, multimedia, interaction method, and text simultaneously, this observation cannot isolate the specific contribution of the keyboard-based interaction component from the accessibility package.

B. Severity of Baseline Barriers and the Value of the Control Condition

Only two of the eight control-group participants (25%) reached concept exposure to all three programming concepts independently in the feature-reduced baseline version (Table VI); the remaining six stopped at some point between Level 1 and Level 4, with the attrition concentrated within the Sequences concept itself, where four of the eight participants stopped at Level 1 or Level 2. This is a substantially more severe barrier than the usability scores alone would suggest: the control-group overall usability score fell from Moderately Good for Sequences (2.81) to Weak for both Loops (1.56) and Conditionals (0.57), but because these scores are computed across all eight participants, including zero-scored non-completions, the numeric decline understates the practical severity captured in Table VI: a quarter of participants, not a numeric average, is arguably the more informative statistic for judging whether the baseline environment is usable at all for this population. The reasons for non-completion recorded in Table VI point to the interaction method and the screen design as the two most frequently observed points of difficulty: four of the six non-completing participants were recorded as unable to complete mouse-based drag-and-drop block placement, and the remaining difficulties involved distinguishing low-contrast blocks and losing track of the task without audio cues. Since both versions differ only in the accessibility elements listed in Table I (screen design, multimedia, interaction method, and text), and the control-group attrition is concentrated at the earliest levels while the experimental group performed well on the same concepts, the pattern is consistent with the removed accessibility package as a whole rather than the increase in conceptual complexity alone being the primary driver of the gap, since conceptual complexity was held constant between conditions. Because the accessible and baseline versions differ in multiple accessibility elements simultaneously (Table I), the specific contribution of the interaction method relative to the other adaptations cannot be isolated from the present design.

C. Learning Outcomes

All eight experimental-group participants showed a positive learning gain, ranging from +5.0 to +6.5 points (M=5.69, SD=0.53), a consistent pattern with no participant falling below the group's lower bound. The two control-independent participants who reached full concept exposure in the baseline version showed smaller gains (+2.5 and +2.0). As established in Section III-D3, this comparison must be read as a conservative estimate of the accessibility effect rather than a representative group comparison: it includes only the two control participants best positioned to learn from the baseline environment (Table VI), while the six control participants who did not reach concept exposure and who, by definition, had markedly lower learning opportunity are absent from the comparison entirely. The true

gap between the accessible and baseline conditions is therefore likely larger than the individual-level gain figures alone suggest.

Relating this to the usability findings in Section V-A, the experimental group's gains remained consistently high (≥+5.0) even for Conditionals, the concept at which usability had declined to Moderately Good rather than Good. This suggests that, within the accessible version, the residual usability decline observed for more complex concepts did not translate into a proportional decline in learning outcome: a Moderately Good usability level for a given concept still supported a learning gain comparable to concepts rated Good. This pattern would benefit from confirmation with a larger and more complexity-differentiated sample, given the descriptive nature of this dataset.

D. Design Tensions Between Playability and Accessibility

The termination-criterion event data provide indirect evidence of one such tension: keyboard navigation speed relative to task completion time. The experimental group's efficiency scores declined from 4.22 (Sequences) to 2.97 (Conditionals) even though effectiveness (task success) remained comparatively higher (4.54 to 3.29), a gap plausibly consistent with the keyboard-based selection method requiring more discrete key presses to construct increasingly complex block arrangements, trading interaction precision and accessibility for the raw speed of a direct drag-and-drop action; as with the other component-level observations in this study, this specific mechanism cannot be isolated from the accessibility package as a whole (Table I). This did not, however, translate into disengagement: experimental-group participants who triggered a proactive check-in continued the session in four of five instances, versus nine of fourteen check-in instances in the control group ending in early termination, suggesting that the accessible version's slower interaction pace was tolerated by children rather than experienced as frustrating enough to prompt withdrawal. This extends the accessibility model of [11] originally validated for a tap-to-select mathematics game to the specifically higher-friction case of block-based programming, indicating that the accessible version's design, including its keyboard-based interaction method, can preserve engagement even where it does not match the raw efficiency of unrestricted pointer-based interaction. A full account of the trade-off between audio feedback and game immersion, and between visual simplification and game attractiveness, requires the qualitative observation notes referenced in Section III-C6 and is not addressed by the quantitative data reported here.

E. Comparison with Prior Work

The overall effectiveness (3.91), efficiency (3.60), and satisfaction (3.91) scores obtained for the accessible version in this study are all lower than the corresponding scores reported for an accessible mathematics serious game for the same population (4.74, 4.55, and 4.92, respectively) [3]. Because both studies applied a broadly similar accessibility model, screen design, multimedia, and language adaptations derived from [11], this gap is more plausibly attributable to the additional interaction barrier specific to block-based programming than to a difference in the underlying accessibility approach: a mathematics game in which children select and tap an answer does not require the sequential, multi-step block construction demanded by sequences, loops, and conditionals, even when both interactions are mediated through an accessible, non-drag-and-drop input method. This is consistent with the distinction drawn in Section I between accessibility barriers that screen design and multimedia adaptations alone can resolve, and the interaction-specific barrier that block-based programming introduces regardless of those adaptations [20]. Relative to the fully non-visual, audio-based approach used for blind children in [5], the present results suggest that the accessible version's overall approach, preserving visual access through a keyboard-based interaction method combined with the other accessibility adaptations in Table I, rather than removing the visual channel entirely, is a viable middle path for the residual-vision case of low vision children, achieving Good overall usability despite the added interaction complexity of block-based programming. This is broadly consistent with assistive-technology approaches for mathematics learning among visually impaired populations, which similarly favour input methods that preserve residual vision where possible [28].

F. Limitations

This study has several limitations. First, the sample size (n=16, split into matched experimental and control groups of 8 participants each) is small; findings are reported descriptively and are not intended to support statistical generalisation. Second, participants were recruited from a single special education school; although experimental and control sessions were scheduled on separate days to minimise cross-group contamination, the sample reflects the specific educational and cultural context of that institution. Third, children with additional cognitive or motor impairments were excluded to reduce confounding, which limits the applicability of the findings to low vision children with co-occurring conditions. Fourth, although both groups received an equal fixed session duration, this does not imply an equal experience within that time: an experimental group participant who completed the core levels early spent the remainder of the session on non-core decorative activities, while a control group participant who struggled to progress may have spent most or all of the session attempting the first core level without success; this qualitative difference in what the time was spent on is a limitation of the equal-duration design. Fifth, the learning-gain comparison reported as individual case data in Table VII for the control-independent subgroup is subject to survivorship bias. Sixth, this study assessed learning gain immediately after the session; it does not assess retention of programming concepts over time. Seventh, beyond parental or guardian consent and the children's right to withdraw described in Section III-B, the study protocol did not include a separate, age-appropriate child assent procedure, nor did it specify formal data-protection arrangements for the screen and camera recordings collected during testing; both are noted here as procedural gaps that future replications of this design, particularly given the involvement of children with disabilities, should address explicitly.

VI. Conclusion and Future Work

In this study, a block-based programming serious game was designed for low-vision children aged seven years old, incorporating accessibility elements such as screen design, multimedia, and a keyboard-based interaction method in place

of the drag-and-drop interaction typically used in block-based programming environments. The prototype was refined through pilot testing and expert heuristic evaluation before user testing. Usability testing and a pre-test/post-test learning outcome assessment were conducted, benchmarked against a feature-reduced baseline version of the game. The accessible version achieved a Good overall usability score (3.80), compared to a Weak overall score (1.65) for the feature-reduced baseline version when computed across the full control group. Only 25% of control-group participants reached independent exposure to all three target concepts in the baseline environment, compared with 100% of experimental-group participants in the accessible version. Experimental-group participants showed a consistent learning gain (M=5.69, SD=0.53, range +5.0 to +6.5), descriptively larger than the gains observed for the two control-group participants who reached concept exposure independently (+2.5 and +2.0), a comparison interpreted conservatively due to survivorship bias in the control-independent subgroup. Accessibility elements such as screen design, multimedia, and interaction method are important in increasing the usability of block-based programming serious games for low vision children. Future research could extend this design to a larger, multi-site sample of low vision children, incorporate a delayed post-test to assess retention of programming concepts, and explore extending the accessibility model to children with additional visual or motor impairments, building on related efforts to extend computational-thinking tools to younger children [27] and to children with other developmental disabilities such as autism spectrum conditions [29].

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Universiti Sains Malaysia for the invaluable support provided throughout the course of this research. This study would not have been possible without their encouragement and contributions.

DECLARATION ON GENERATIVE AI-ASSISTED TOOLS

Generative AI tools were used for language editing, formatting assistance, and organisation of the manuscript during preparation. The authors critically reviewed, edited, and approved the study design, data collection, analysis, interpretation, references, and final manuscript, and take full responsibility for the scientific content.

REFERENCES

[1] Y. Zhonggen, "A Meta-Analysis of Use of Serious Games in Education over a Decade," *International Journal of Computer Games Technology*, vol. 2019, Article ID 4797032, 8 pages, 2019, doi: 10.1155/2019/4797032.

[2] L. Sun and C. E. Pan, "Effects of the Application of Information Technology to E-Book Learning on Learning Motivation and Effectiveness," *Frontiers in Psychology*, vol. 12, Article 752303, 2021, doi: 10.3389/fpsyg.2021.752303.

[3] N. I. Othman, H. Mohamed, and N. A. M. Zin, "Testing the Usability of Serious Game for Low Vision Children," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 8, pp. 310–317, 2023, doi: 10.14569/IJACSA.2023.0140834.

[4] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch Programming Language and Environment," *ACM Trans. Comput. Educ.*, vol. 10, no. 4, pp. 1–15, 2010, doi: 10.1145/1868358.1868363.

[5] L. R. Milne and R. E. Ladner, "Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments," in *Proc. 2018 CHI Conf. Human Factors in Computing Systems*, 2018, pp. 1–10.

[6] A. Donker and P. Reitsma, "Usability testing with young children," *Proc. 2004 Conf. Interact. Des. Child. Build. a Community, IDC 2004*, pp. 43–48, 2004, doi: 10.1145/1017833.1017839.

[7] M. M. Ali-Shahid and S. Sulaiman, "A case study on reliability and usability testing of a Web portal," *2015 9th Malaysian Softw. Eng. Conf. MySEC 2015*, pp. 31–36, 2016, doi: 10.1109/MySEC.2015.7475191.

[8] N. I. Othman, N. A. M. Zin, and H. Mohamed, "Play-Centric Designing of a Serious Game Prototype for Low Vision Children," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 5, pp. 199–205, 2020, doi: 10.14569/IJACSA.2020.0110528.

[9] H. Y. Abuaddous, M. Z. Jali, and N. Basir, "Web Accessibility Challenges," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 10, 2016, doi: 10.14569/IJACSA.2016.071023.

[10] K. K. Allah, N. A. Ismail, L. Hasan, and W. Y. Leng, "Usability Evaluation of Web Search User Interfaces from the Elderly Perspective," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 12, pp. 647–657, 2021, doi: 10.14569/IJACSA.2021.0121281.

[11] N. I. Othman, H. Mohamed, and N. A. M. Zin, "Serious Games Accessibility Design Model for Low-Vision Children," *Advances in Human-Computer Interaction*, vol. 2023, Article ID 9528294, 2023, doi: 10.1155/2023/9528294.

[12] A. Stefik, W. Allee, G. Contreras, T. Kluthe, A. Hoffman, B. Blaser, and R. Ladner, "Accessible to Whom? Bringing Accessibility to Blocks," in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, 2024, pp. 1286–1292, doi: 10.1145/3626252.3630770.

[13] A. Mountapmbeme and S. Ludi, "Designing Accessible Block-Based Programming Environments for Persons with Visual Impairments," *Universal Access in the Information Society*, vol. 24, no. 3, pp. 2351–2382, 2025, doi: 10.1007/s10209-025-01200-y.

[14] K. Teng and G. K. W. K. Chung, "Measuring Children's Computational Thinking and Problem-Solving in a Block-Based Programming Game," *Education Sciences*, vol. 15, no. 1, Article 51, 2025, doi: 10.3390/educsci15010051.

[15] L. Wöbbekind, K. Lorberg, and T. Mandl, "Emma Stop That, It's My Turn Now — Comparing Peer Tutoring and Thinking Aloud for Usability-Testing with Children in a School Setting," in *Proc. Mensch und Computer 2023*, 2023, pp. 442–447, doi: 10.1145/3603555.3608542.

[16] F. Lehnert, S. Doublet, and G. Sim, "Designing Usability/User Experience Heuristics to Evaluate e-Assessments Administered to Children," *Computer Standards & Interfaces*, vol. 92, Article 103933, 2024, doi: 10.1016/j.csi.2024.103933.

[17] R. Wang, L. Zeng, X. Zhang, S. Mondal, and Y. Zhao, "Understanding How Low Vision People Read Using Eye Tracking," in *Proc. 2023 CHI Conf. Human Factors in Computing Systems*, 2023, pp. 1–17, doi: 10.1145/3544548.3581213.

[18] Y. Wang, Y. Zhao, and Y.-S. Kim, "How Do Low-Vision Individuals Experience Information Visualization?," in *Proc. 2024 CHI Conf. Human Factors in Computing Systems*, 2024, Article 43, doi: 10.1145/3613904.3642188.

[19] R. Cossovich, S. Hodges, J. Kang, and A. Girouard, "Co-designing New Keyboard and Mouse Solutions with People Living with Motor Impairments," in *Proc. 25th Int. ACM SIGACCESS Conf. Computers and Accessibility*, 2023, Article 98, doi: 10.1145/3597638.3614549.

[20] R. H. Mollik, W. Aljedaani, and S. Ludi, "Understanding Design and Accessibility Issues in Block-Based Programming for People with Disabilities: A Literature Review," *Universal Access in the Information Society*, vol. 25, no. 3, Article 94, 2026, doi: 10.1007/s10209-026-01351-6.

[21] K. Żyła, K. Chwaleba, and D. Choma, "Evaluating Usability and Accessibility of Visual Programming Tools for Novice Programmers—The Case of App Inventor, Scratch, and StarLogo," *Applied Sciences*, vol. 14, no. 21, Article 9887, 2024, doi: 10.3390/app14219887.

[22] M. S. Alotaibi and M. B. Garcia, "Game-Based Learning in Early Childhood Education: A Systematic Review and Meta-Analysis," *Frontiers in Psychology*, vol. 15, Article 1307881, 2024, doi: 10.3389/fpsyg.2024.1307881.

[23] M. Spinoso, M. Benassi, N. Mazzoni, M. Orsoni, L. Stefanutti, P. Anselmi, D. de Chiusole, A. Bacherini, I. Pierluigi, S. Garofalo, G. Balboni, and S. Giovagnoli, "Psychometric Properties of the Usability and Acceptability Scale (UAS) for Evaluating Digital Tools in Children and Adolescent Users," *Frontiers in Psychology*, vol. 17, Article 1702085, 2026, doi: 10.3389/fpsyg.2026.1702085.

[24] T. Tene, D. F. Vique López, P. E. Valverde Aguirre, N. I. Cabezas Oviedo, C. Vacacela Gomez, and S. Bellucci, "A Systematic Review of Serious Games as Tools for STEM Education," *Frontiers in Education*, vol. 10, Article 1432982, 2025, doi: 10.3389/feduc.2025.1432982.

[25] S. W. Gundersen and G. Lampropoulos, "Using Serious Games and Digital Games to Improve Students' Computational Thinking and Programming Skills in K-12 Education: A Systematic Literature Review," *Technologies*, vol. 13, no. 3, Article 113, 2025, doi: 10.3390/technologies13030113.

[26] X. Wang, M. Cheng, and X. Li, "Teaching and Learning Computational Thinking Through Game-Based Learning: A Systematic Review," *Journal of Educational Computing Research*, 2023, doi: 10.1177/07356331231180951.

[27] G. P. Perez Valdes, O. Boude Figueredo, and A. D. Vargas Sanchez, "Integrating Computational Thinking in Children Aged 3 to 6: Challenges and Opportunities in Early Childhood Education," *Frontiers in Education*, vol. 10, Article 1535135, 2025, doi: 10.3389/feduc.2025.1535135.

[28] M. Shoaib, D. Fitzpatrick, and I. Pitt, "Assistive Technology-Based Solutions in Learning Mathematics for Visually-Impaired People: Exploring Issues, Challenges and Opportunities," *Multimedia Tools and Applications*, vol. 82, pp. 46153–46184, 2023, doi: 10.1007/s11042-023-17409-z.

[29] K. F. Carhuaz Malpartida and K. Rios da Hora Rodrigues, "Building Serious Games to Exercise Computational Thinking: Initial Evaluation with Teachers of Children on the Autism Spectrum," *Journal on Interactive Systems*, vol. 16, no. 1, 2025, doi: 10.5753/jis.2025.4492.

[30] ISO 9241-11:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*, International Organization for Standardization, Geneva, 1998. (Superseded by ISO 9241-11:2018)

[31] University of Pittsburgh Department of Ophthalmology, "What is Vision Impairment?" [Online]. Available: https://www.ophthalmology.pitt.edu/vision-impairment/what-vision-impairment. [Accessed: Sep. 10, 2026].

Hybridization of FA-YOLOv8 Model with Zebra Optimization Algorithm for Improving Traffic Light Detection and Tracking Under Low-Light Conditions

Adhwa Salemi, Muhammad Arif Mohamad*
Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Malaysia

***Abstract*—Traffic-light detection and tracking under low-light conditions remain difficult because signal heads can be small, dim, blurred, low in contrast, or affected by glare and reflections. This study develops a hybridization of Firefly Algorithm–You Only Look Once version 8 (FA-YOLOv8) with the Zebra Optimization Algorithm (ZOA) for offline analysis of recorded dashcam video under rainy, night-time, and cloudy conditions. YOLOv8 detects traffic-light candidates, FA performs attraction-based coordinate refinement and supports tracking continuity, and ZOA acts as an enhancement layer that adaptively searches FA control settings rather than replacing FA as the refinement mechanism. A controlled within-dataset comparison uses 300 real-world dashcam frames containing approximately 900 annotated traffic-light instances; the same annotated sequences support parameter adaptation and the baseline-versus-hybrid comparison, so the reported results characterize within-dataset behavior rather than held-out generalization. Compared with baseline FA-YOLOv8, MOTP increases from 31.84% to 37.59% under rainy conditions and from 49.47% to 72.50% under cloudy conditions. At night, Recall increases from 54.70% to 80.34% and F1-Score from 61.73% to 75.56%. Tracking continuity also improves in the reported point estimates: IDF1 rises from 72/90/85% to 82/94/90% for rainy/night/cloudy conditions, while identity switches fall from 95/18/45 to 48/8/22. These observations show that ZOA-based enhancement changes FA search behavior and is associated with improvements in several localization and identity-tracking measures within the evaluated sequences, although detection changes remain condition-dependent.**

Keywords*—*Traffic light detection; traffic light tracking; YOLOv8; firefly algorithm; zebra optimization algorithm; low-light conditions; adaptive parameter optimization; identity tracking

I. Introduction

Traffic-light perception requires reliable detection, localization, and tracking across consecutive road frames [1]. Under low-light conditions, weak contrast, glare, reflections, and rain blur can distort candidate locations and interrupt temporal continuity. These effects motivate a framework that can refine detected coordinates while adapting its search behavior to the illumination condition rather than relying on fixed optimization parameters.

The central problem addressed here is not simply low-light detection; it is the parameter sensitivity of FA when it is used to refine and track YOLOv8 detections. FA offers a useful attraction-based local search mechanism, but its convergence and exploration–exploitation behavior depend on control parameters [2]. A fixed configuration can therefore behave differently when the scene changes from cloudy to night-time or rainy conditions.

YOLOv8 is retained as the detection backbone and is not the main novelty of this study. The contribution lies in the optimization layer placed around the detector output: FA refines candidate coordinates, HSV analysis supports traffic-light state recognition, and ZOA adapts the FA control parameters to the current search difficulty.

Accordingly, the proposed method hybridizes FA-YOLOv8 with ZOA rather than replacing FA with another tracker. ZOA is introduced specifically as an enhancement layer for the existing FA-YOLOv8 framework: FA remains responsible for attraction-based coordinate refinement and temporal continuity, while ZOA searches the FA control settings, including the maximum generation setting and the parameters alpha, beta, and gamma. This complementary division targets FA parameter sensitivity while preserving the original refinement mechanism.

The contributions are fourfold. First, a FA-YOLOv8-ZOA hybridization is formulated for low-light traffic-light detection and tracking. Second, ZOA is used to enhance the existing FA-YOLOv8 framework through adaptive FA parameter search across rainy, night-time, and cloudy conditions. Third, the baseline FA-YOLOv8 and proposed hybrid are evaluated using detection/localization and identity-tracking metrics. Fourth, the analysis distinguishes condition-dependent changes and explicitly limits its conclusions to the offline, within-dataset experiment rather than claiming real-time deployment, independent generalization, or universal optimizer superiority.

II. Related Work

A. Low-Light Traffic-Light Perception

Low-light traffic-light detection is difficult because illumination changes simultaneously affect color, contrast, and object boundaries [3]. Deep detectors improve robustness over handcrafted methods, yet small and distant signals remain vulnerable to missed detections and unstable boxes when glare, rain reflections, or weak illumination suppress visual evidence [4]. These studies motivate robust perception, but they do not directly address the parameter sensitivity of an optimization-based refinement layer placed after detector output.

Tracking adds a temporal requirement: a correct detection must remain spatially stable and preserve its identity across frames [5]. Existing low-light tracking work commonly

*Corresponding author.

focuses on detector design, preprocessing, or tracking architecture [5], [6]. In contrast, the present study focuses on a narrower problem inherited from the baseline FA-YOLOv8 framework: how to enhance FA search behavior when fixed control settings become less suitable as illumination conditions change.

B. Firefly Algorithm as the Baseline Refinement Mechanism

The Firefly Algorithm (FA) is a swarm-based optimizer in which weaker candidate solutions move toward brighter, higher-fitness solutions [7]. Its attraction mechanism has previously been applied to object tracking [8]. In the baseline FA-YOLOv8 framework, this behavior is used to refine detected traffic-light coordinates and support continuity across frames [9]. The main limitation relevant to this study is not the absence of a refinement mechanism, but the sensitivity of FA convergence and search behavior to fixed control settings.

In the baseline FA-YOLOv8 formulation used in this study, each firefly represents a candidate traffic-light coordinate, and its brightness represents the quality of that candidate [9]. The implemented light-intensity relationship is expressed as:

$$I_i = I_0 e^{(-\gamma * fitness_i)} \quad (1)$$

where, I_i is the updated light intensity of candidate i, I_0 is the initial intensity, γ is the light-absorption coefficient, and $fitness_i$ represents the candidate quality. A weaker candidate moves toward a stronger candidate using the implemented update:

$$x_i(t+1) = x_i(t) + \beta\left[x_j(t) - x_i(t)\right] + \alpha * L() \quad (2)$$

where, alpha controls randomized step size, beta controls attraction toward the better candidate, and L() denotes the stochastic Lévy-flight component used in the implementation. The process iterates until the termination criterion is met and the best coordinate is returned. These parameters are effective only when appropriately tuned; static settings can be suboptimal when the illumination condition changes [10].

C. Zebra Optimization Algorithm for FA Parameter Adaptation

ZOA models zebra foraging and defensive behavior through complementary search phases [11]. In this work, ZOA is not used as an independent replacement for FA and is not treated as a competing tracker. Instead, it is embedded as an enhancement mechanism that searches FA control settings and retains settings that improve the implemented FA fitness. This design is consistent with the thesis objective of improving the existing FA-YOLOv8 framework rather than replacing its attraction-based refinement stage.

This role separation defines the hybridization studied here: FA performs localized coordinate refinement, whereas ZOA modifies how FA searches by adapting its control settings [12]. Prior work establishes the individual usefulness of FA and ZOA, but does not directly address this specific use of ZOA as a parameter-enhancement layer for FA-YOLOv8 in low-light traffic-light tracking. The proposed design therefore targets parameter sensitivity in the baseline framework without changing the YOLOv8 detector itself.

III. Proposed Hybridization of FA-YOLOv8 with ZOA

As illustrated in Fig. 1, the processing pipeline begins with prerecorded dashcam frames and YOLOv8 candidate detections. HSV-based pixel analysis supports traffic-light state recognition, after which FA refines candidate coordinates. For each processed frame containing traffic-light candidates, the baseline FA configuration is evaluated first, ZOA searches alternative FA control settings, and the selected settings are used for the final FA refinement pass. The refined detections are then evaluated using detection, localization, and identity-tracking measures. Thus, the adaptive feedback shown in Fig. 1 is applied during offline frame processing rather than as a once-per-sequence global tuning stage.

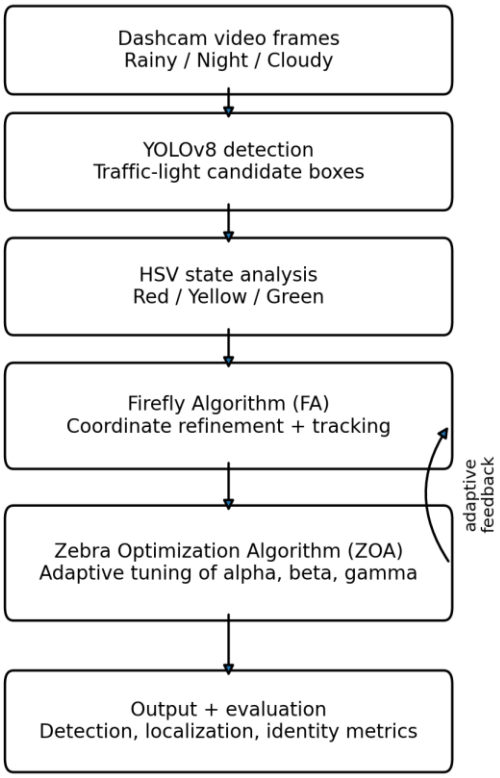


Fig. 1. Processing flow of the proposed FA-YOLOv8-ZOA hybridization.

A. YOLOv8 Detection and HSV State Analysis

YOLOv8 is responsible for candidate localization, class prediction, and confidence information. HSV analysis is used within the detected region to help distinguish the red, yellow, and green traffic-light states. The metaheuristic stages operate on the detected candidates; they do not replace YOLOv8 feature extraction or class prediction.

B. FA Coordinate Refinement and ZOA Parameter Adaptation

FA begins with candidate positions derived from the detector output. Candidate fitness is evaluated, brightness is updated, and weaker fireflies move toward better candidates according to the attraction and stochastic movement terms in Eq. (1) and (2). ZOA operates at the parameter level: candidate settings are sampled within the implementation bounds, evaluated by running FA, and retained when the resulting FA

fitness improves. The selected settings are then returned to FA for the final coordinate-refinement pass. This preserves FA as the refinement mechanism while using ZOA only to enhance its search configuration.

Operationally, the implementation searches a parameter vector containing maximum generations together with alpha, beta, and gamma. The implemented fitness is F = max(0, N - 0.1D), where N is the number of traffic-light candidates, and D is the sum of pairwise Euclidean distances among candidate centers. The optimizer therefore favors configurations that preserve candidate count while reducing spatial dispersion; Precision, Recall, F1-Score, IDF1, and IDSW are evaluation measures reported after processing rather than weighted terms inside the ZOA fitness function. The implementation searches max generation in [50, 200], alpha in [0.1, 0.5], beta in [0.5, 2.0], and gamma in [0.1, 1.0]. The archived code uses a ZOA population size of 1 and one ZOA update iteration per processed frame. The baseline FA initialization uses five fireflies with max generation = 100, alpha = 0.2, beta = 1.0, and gamma = 0.2. These implementation details define the parameter-enhancement mechanism evaluated in this study.

IV. Experimental Setup

A. Dataset and Selection Criteria

The experimental evaluation uses prerecorded real-world traffic-light video captured with a vehicle-mounted dashcam and processed offline. The data contain 300 selected frames and approximately 900 annotated traffic-light instances, divided equally across rainy, night-time, and cloudy conditions, as summarized in Table I. The same annotated sequences are used as feedback during ZOA adaptation of the existing FA-YOLOv8 outputs and for the subsequent baseline-versus-hybrid measurements; YOLOv8 is not retrained during ZOA adaptation. Accordingly, this study is a controlled within-dataset comparative evaluation of the incremental effect of adding ZOA-based enhancement to FA-YOLOv8 rather than an independent held-out generalization experiment. Because this study evaluates offline recorded video, online throughput, FPS, and embedded real-time deployment are not defined as primary experimental objectives.

TABLE I. Dataset Distribution Under Selected Low-Light Conditions

Condition	Frames	Annotated TLs	Primary Challenge
Rainy	100	300	Reflection and rain blur
Night	100	300	Low illumination
Cloudy	100	300	Inconsistent lighting
Total	300	900	Controlled within-dataset comparison

The dataset was filtered according to the predefined selection criteria summarized in Table I: traffic lights had to be visible from a direct frontal view, motion blur had to remain limited, object size had to be detectable, and severely overexposed, extremely dark, fully occluded, or excessively blurred cases were excluded. These criteria define the controlled scope of the reported results and should be considered when interpreting performance outside the evaluated low-light scenes.

B. Evaluation Metrics

Performance is evaluated with MOTP, Precision, Recall, and F1-Score to describe detection and localization behavior. Because the proposed method also targets temporal continuity, the analysis includes IDF1, IDSW, Mostly Tracked (MT), Partially Tracked (PT), and Mostly Lost (ML). These metrics are used to compare the same baseline and enhanced framework under the three conditions listed in Table I.

The principal experimental control is baseline FA-YOLOv8 versus the proposed FA-YOLOv8-ZOA framework because the research objective is to determine the incremental effect of using ZOA to enhance the previously developed FA-YOLOv8 model. The broader study also contains supporting comparisons with other optimization-based YOLOv8 variants, but the present article does not make a universal claim that ZOA is superior to every possible parameter-search strategy. Its central question is whether the addition of ZOA-based parameter enhancement changes the performance of FA-YOLOv8 under the tested low-light conditions.

V. Results and Discussion

A. Adaptive FA Parameter Behavior

Table II shows that ZOA returns different mean FA parameter values for the three illumination conditions, supporting the premise that one fixed FA configuration is not equally suitable for every scene type. Rainy conditions require the largest mean maximum generation, while the night-time subset uses a lower mean alpha and a higher mean gamma. These differences indicate condition-dependent search behavior rather than a single global parameter setting.

TABLE II. Mean Adaptive FA Parameters Generated By ZOA

Condition	Max Gen	α	β	γ
Rainy	155.83	0.25	1.56	0.47
Night	139.08	0.15	1.42	0.59
Cloudy	125.25	0.21	1.33	0.33

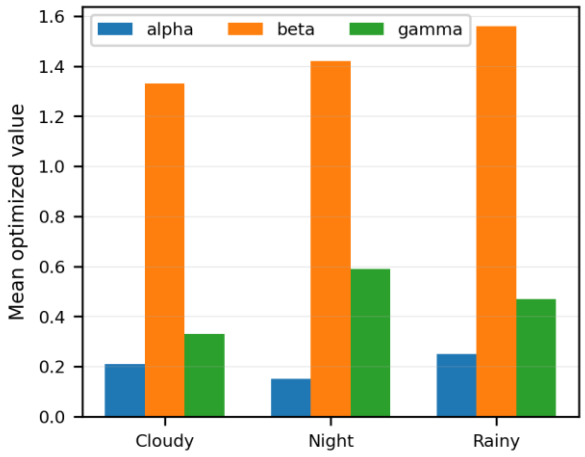


Fig. 2. Mean FA parameter values produced by adaptive ZOA tuning.

The parameter analysis in Table II and Fig. 2 provides mechanism-level evidence for the proposed hybridization. In

baseline FA-YOLOv8, the control settings are relatively static, whereas the hybrid framework changes them through ZOA. Within the tested data, this enhancement is associated with condition-dependent FA search behavior and changes in localization and tracking continuity.

B. Detection and Localization Results

As shown in Table III, under rainy conditions, MOTP increases from 31.84% to 37.59%, while Precision increases from 73.94% to 100%. Recall decreases from 25.45% to 22.92% and F1-Score decreases slightly from 37.83% to 37.29%. The 100% Precision value should therefore not be interpreted as perfect overall detection: it occurs together with low Recall, indicating a conservative operating point in which the reported detections contain no counted false positives while a substantial proportion of annotated traffic lights remain undetected. The rainy result therefore indicates improved localization precision and fewer false detections, but not a uniform improvement in detection completeness.

TABLE III. DETECTION AND LOCALIZATION RESULTS

Metrics	Conditions (FA-YOLOv8)			Conditions (Hybrid)		
	Rainy	Night	Cloudy	Rainy	Night	Cloudy
MOTP (%)	31.84	32.84	49.47	37.59	39.74	72.50
Precision (%)	73.94	71.11	100.00	100.00	71.21	100.00
Recall (%)	25.45	54.70	25.57	22.92	80.34	35.25
F1-Score (%)	37.83	61.73	40.72	37.29	75.56	52.15

The night-time results in Table III show a different pattern. MOTP increases from 32.84% to 39.74%; Recall rises from 54.70% to 80.34%; and F1-Score rises from 61.73% to 75.56%, while Precision remains nearly unchanged at 71.11% versus 71.21%. These changes indicate stronger detection completeness together with improved localization behavior under night-time illumination.

Cloudy scenes in Table III provide the strongest localization gain: MOTP increases from 49.47% to 72.50%, Recall rises from 25.57% to 35.25%, and F1-Score increases from 40.72% to 52.15%, while Precision remains at 100%. As in the rainy result, 100% Precision should be read together with the relatively low Recall rather than as evidence of perfect detection. Across the three conditions, the results show improvements in several, but not all, detection and localization measures.

C. Tracking Continuity and Identity Consistency

Table IV and Fig. 3 show that identity continuity improves in the reported point estimates for all three evaluated environments. IDF1 increases from 72% to 82% in rain, 90% to 94% at night, and 85% to 90% in cloudy scenes. Identity switches fall from 95 to 48, 18 to 8, and 45 to 22, respectively. MT also increases, while ML decreases across all three conditions.

These tracking results form the clearest observed pattern in the current experiment: after ZOA-based enhancement, the reported sequences show fewer identity switches and higher IDF1. They complement the detection metrics because temporal association can improve even when a frame-level metric such as Recall or F1-Score does not improve. The interpretation is restricted to the evaluated runs and does not claim statistical variability across repeated stochastic trials.

TABLE IV. TRACKING AND IDENTITY CONSISTENCY RESULTS

Conditions	IDF1 (%)	IDSW	MT (%)	PT (%)	ML (%)
Rainy	72 -> 82	95 -> 48	45 -> 58	40 -> 34	15 -> 8
Night	90 -> 94	18 -> 8	80 -> 88	18 -> 11	2 -> 1
Cloudy	85 -> 90	45 -> 22	65 -> 74	30 -> 23	5 -> 3

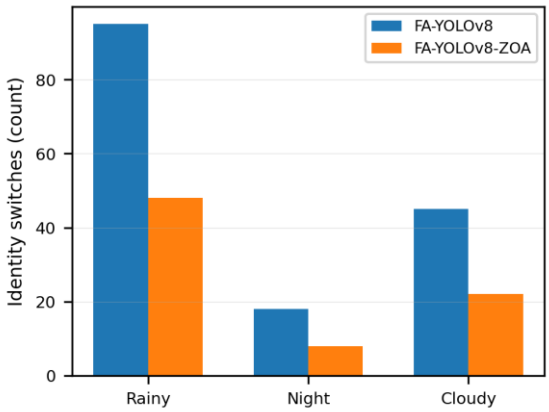


Fig. 3. Reduction in identity switches after YOLOv8 was hybridized with ZOA.

D. Qualitative Detection and Tracking Behaviour

The representative outputs in Fig. 4 are consistent with the quantitative observations. Baseline FA-YOLOv8 detects traffic-light candidates in all three environments but shows moderate coordinate and continuity stability in difficult frames. After ZOA-based parameter enhancement is introduced, the illustrated outputs show more stable localization and stronger identity preservation, particularly under night-time and cloudy illumination.



Fig. 4. Representative inputs, baseline FA-YOLOv8 outputs, and proposed FA-YOLOv8-ZOA outputs for cloudy, night, and rainy scenes.

E. Extension Beyond the Baseline FA-YOLOv8 Study

This work extends the previously published FA-YOLOv8 framework [2]. The earlier study established the feasibility of combining FA with YOLOv8 for traffic-light detection; the present study adds ZOA as an enhancement layer for adaptive FA parameter search, condition-specific parameter analysis, and a broader evaluation of localization and identity continuity [13].

Unlike approaches that primarily modify detector architecture, enhancement preprocessing, or multi-camera fusion [1], [2], the proposed contribution operates at the optimization/refinement stage. Its purpose is not to replace FA, but to enhance how the FA component searches and refines coordinates after YOLOv8 has produced candidate detections.

VI. Conclusion

This study hybridizes FA-YOLOv8 with the Zebra Optimization Algorithm as an enhancement mechanism for offline traffic-light detection and tracking under selected low-light conditions. YOLOv8 supplies candidate detections, FA remains responsible for attraction-based localization refinement, and ZOA searches FA control settings to adapt the refinement behavior to illumination-dependent difficulty.

Within the controlled same-sequence comparison, the hybrid shows favorable changes in several localization and tracking point estimates. Rainy MOTP and Precision increase, cloudy MOTP increases substantially, night-time Recall and F1-Score improve, and the reported identity metrics show higher IDF1 and fewer identity switches in all three tested conditions. At the same time, rainy Recall and F1-Score do not improve, and the 100% Precision cases coexist with low Recall. These findings therefore support a narrower conclusion: ZOA-based enhancement modifies FA search behavior and is associated with improvements in selected metrics within the evaluated offline sequences; they do not establish real-time deployment performance, universal optimizer superiority, or independent generalization to unseen traffic scenes.

Future work should follow the constraints already identified in this study by expanding the dataset to more diverse intersections, camera viewpoints, and more severe visibility conditions such as fog, heavy rain, extreme glare, tunnel exits, overexposure, and stronger occlusion. If the framework is later adapted for online or vehicle-based deployment, computational efficiency, hardware-specific runtime, and embedded implementation should then be evaluated explicitly. Additional road-object classes and broader traffic-scene coverage can also be investigated without changing the central role of ZOA as an enhancement to FA-YOLOv8.

Acknowledgment

The authors acknowledge the Ministry of Higher Education Malaysia (MOHE) for funding under the Fundamental Research Grant Scheme-Early Career Researcher (FRGS-EC) FRGS-EC/1/2024/ICT02/UMPSA/02/1 and offer heartfelt thanks to the guidance of the Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah.

Declarations

Declaration on Generative AI. OpenAI ChatGPT was used to assist with manuscript organization, drafting, and language refinement. The authors critically reviewed the AI-assisted text and remain fully responsible for the scientific ideas, experimental values, interpretations, references, originality, and final manuscript.

Thesis derivation. This manuscript is derived from and substantially reorganizes the experimental work reported in the first author's Master of Science thesis. The journal article focuses on the FA-YOLOv8-to-FA-YOLOv8-ZOA hybridization, adaptive parameter behavior, and condition-wise detection/tracking evaluation.

References

[1] A. Salemi and M. A. Mohamad, "Critical Review of Object Detection Techniques for Traffic Light Detection in Intelligent Transportation Systems." doi: https://dx.doi.org/10.14569/IJACSA.2025.0161072.

[2] A. Salemi, M. A. Mohamad, and Z. Musa, "Hybrid Firefly Algorithm and You Only Look Once v8 Framework (FA-YOLOv8) for Traffic Light Detection," 2026. doi: 10.1109/icsecs65227.2025.11279255.

[3] P. Zhong et al., "Low-Light Object Tracking: A Benchmark," Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.11463

[4] A. Yi and N. Anantrasirichai, "A Comprehensive Study of Object Tracking in Low-Light Environments," Sensors, vol. 24, no. 13, Jul. 2024, doi: 10.3390/s24134359.

[5] A. Albouchi, S. Messaoud, S. Bouaafia, M. A. Hajjaji, and A. Mtibaa, "Implementation of an improved multi-object detection, tracking, and counting for autonomous driving," Multimed. Tools Appl., vol. 83, no. 18, 2024, doi: 10.1007/s11042-023-17444-w.

[6] K. R. Ranipa and K. R. Bhatt, "Illumination Condition Effect on Object Tracking: A Review," European Journal of Academic Essays ISSN, vol. 1, no. 11online, pp. 27–31, 2014, [Online]. Available: www.euroessays.org

[7] V. Kumar and D. Kumar, "A Systematic Review on Firefly Algorithm: Past, Present, and Future," Archives of Computational Methods in Engineering, vol. 28, no. 4, 2021, doi: 10.1007/s11831-020-09498-y.

[8] T. Kumar, K. Kumar, K. Dhanuk, A. K. Bhargava, and M. K. Sharma, "Firefly algorithm driven intelligent system for multi-objective multimodal transportation optimization," Systems and Soft Computing, vol. 7, 2025, doi: 10.1016/j.sasc.2025.200424.

[9] F. S. Hsieh, "Comparison of a Hybrid Firefly–Particle Swarm Optimization Algorithm with Six Hybrid Firefly–Differential Evolution Algorithms and an Effective Cost-Saving Allocation Method for Ridesharing Recommendation Systems," Electronics (Switzerland), vol. 13, no. 2, 2024, doi: 10.3390/electronics13020324.

[10] M. K. Hoang, T. A. Le, K.-X. Thuc, T. Van Luyen, X.-S. Yang, and D. W. K. Ng, "Firefly Algorithm for Movable Antenna Arrays," IEEE Wireless Communications Letters, pp. 1–1, Sep. 2024, doi: 10.1109/lwc.2024.3456899.

[11] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3172789.

[12] M. Eslami, H. Shareef, and M. Khajehzadeh, "Firefly algorithm and pattern search hybridized for global optimization," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013. doi: 10.1007/978-3-642-39482-9_20.

[13] W. He, S. Ma, C. Wang, and Q. Zhang, "Improve zebra optimization algorithm with adaptive oscillation weight and golden sine operator," Feb. 05, 2024. doi: 10.21203/rs.3.rs-3820826/v1.

Enhanced Explainable Multi-Agent Generative Artificial Intelligence Framework for Personalized Education

Hesham M. A. Abdullah[1], Elham Alzain[2]*
Al-Fayha Private College, Jubail, Saudi Arabia[1]
Applied College, King Faisal University, Al-Ahsa, 31982, Saudi Arabia[2]

***Abstract*—Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) have enabled personalized education via intelligent tutoring, adaptive learning, automated assessments, and personalized feedback. Nevertheless, educational AI systems are still confronted with challenges, including high computational complexity, long inference time, unreliable content generation, limited explainability, and ineffective multi-agent orchestration. To this end, this study proposes LITE-Trust, a Lightweight Trust-Aware Explainable Multi-Agent Generative Artificial Intelligence Framework for personalized education, which consists of learner profiling, Knowledge State Estimation (KSE), the Educational Library and Readers Optimization (ELRA) algorithm, relevance-based selective agent orchestration, trust evaluation, and an Explainability Engine. The ELRA algorithm is applied to optimize learner-specific educational configurations by taking learning requirements, trust, and computational efficiency into account, and selective orchestration is conducted to reduce computational overhead by only activating relevant agents. Experiments on the EdNet and Open University Learning Analytics Dataset (OULAD) demonstrate that LITE-Trust achieves 10.5/9.5ms latency, 80.5/81.3 MB memory, and 1.92 M parameters, which is computationally more efficient than the compared methods. In addition, LITE-Trust attains 95.0%/94.6% fidelity, 94.1%/93.4% explanation quality, and 92.6%/92.1% interpretability on the two datasets. Besides, the ELRA algorithm significantly outperforms existing algorithms in terms of various metrics. Overall, the experimental results verify that LITE-Trust can effectively balance personalization, learning quality, trust, explainability, and computational efficiency, thereby facilitating the practical application of lightweight and trustworthy GenAI systems in personalized education.**

Keywords*—Generative artificial intelligence; artificial intelligence in education; personalized learning; multi-agent systems; explainable artificial intelligence; trustworthy AI; educational optimization; lightweight artificial intelligence; library and readers algorithm; learning analytics

I. Introduction

Artificial Intelligence in Education (AIED) has evolved from rule-based educational systems into Intelligent Tutoring Systems (ITSs), personalized learning, Generative AI, and LLMs [1, 2]. These advances allow the creation of content, recommendations, and feedback and offer tutoring services to learners individually or in groups. Still, the growing complexity of AIED and the use of large-scale models and many agents have made education-related artificial intelligence challenging to develop and control.

Current educational AI systems suffer from several limitations. Large-scale transformer-based models increase the computational complexity, memory consumption, inference time, and cost of deploying these systems. Multi-agent educational systems execute multiple agents without considering relevance to the current educational context, thus introducing unnecessary costs [3]. In addition, educational tutoring systems currently lack reliable mechanisms to assess trust and hallucination, potentially propagating unsubstantiated or factually incorrect information to the learner. Finally, these systems have limited explainability, as neither the learners nor instructors are often privy to the decision-making process of the AI. Most importantly, existing optimization methods are typically single-point, only considering one optimization criterion (e.g., prediction accuracy or learning path optimization) rather than jointly optimizing prediction, learning path adaptation, trust, explainability, agent coordination, and computational cost.

These limitations motivate this work by highlighting several research challenges, including single-agent, static multi-agent, limited trust-aware optimization, insufficient explainability, high computational cost, the absence of lightweight educational AI, and limited educational optimization. Table I explains the comparison of proposed and existing approaches.

TABLE I. Comparison of Existing Approaches and LITE-Trust

Methods	Personalization	Multi-Agent	Trust	XAI	Lightweight	Optimization
Existing method	✓	Partial	✗	Partial	✗	Partial
LITE-Trust	✓	✓	✓	✓	✓	✓

The proposed LITE-Trust is a lightweight multi-agent educational AI framework that comprises 1) learner profiling, 2) KSE, 3) ELRA, 4) dynamic agent orchestration, 5) trust evaluation, and 6) explainability. The ELRA optimizes learner-specific educational configurations while relevance-based pruning minimizes unnecessary agent execution. On the other hand, trust evaluation enhances the credibility of produced

*Corresponding author.

educational content, and the explainability engine offers learner-specific rationales. The major contributions include the following:

- A lightweight multi-agent educational AI framework that incorporates personalization, optimization, trust, and explainability features.
- The ELRA algorithm for personalized education optimization.
- A trust-aware evaluation method for enhancing the reliability of generated educational content.
- An explainability engine that enables transparent personalization decisions.
- A dynamic agent orchestration technique that minimizes computational overhead while maximizing the accuracy of personalization.
- A comprehensive evaluation of the performance in terms of learning, trust, explainability, optimization, and computational efficiency.

The rest of this study is structured as follows: Section II describes the related work. Section III presents the LITE-Trust framework. Subsequently, Section IV provides an experimental analysis and discussion. Finally, Section V concludes this study and identifies limitations and future directions.

II. Related Works

A. AI and Generative AI in Education

AI and Generative AI methods efficiently enhance collaborative learning, AI-enabled teaching, and lesson preparation in educational systems. The following presents a survey of AI and Generative AI in educational systems. Alyoussef et al. [4] and Algerafi et al. [5] present a technology acceptance model-based AI adoption architecture for cooperative learning. The system incorporates perceived accessibility, recognized usefulness, engagement factors, and trust attitude to evaluate AI adoption. To evaluate the relationships between these factors, this the study uses structural equation modeling. The system enhances students' behavioural intention and AI-supported technologies in learning environments. Oh et al. (2025) present the AI-based MATH41 system for mathematics teacher education. This approach is employed for generating teaching materials for students and teaching with the aid of Generative AI and MATH41 methods. Here, the learning materials are prepared by the system, which helps to enhance lesson planning skills and AI-enabled learning.

Overall, the discussed research highlights the potential of using AI and GenAI for enhancing engagement, adoption, and content development; however, they are focused on different aspects of using these technologies. The adoption-centric studies explore behavioural and psychological aspects, while GenAI research mainly concentrates on content and teaching materials. Hence, there is a gap in the studies, which implies the need for further investigations. Specifically, there is a need for studies exploring ways of combining learner state adaptation and using GenAI for generating decisions with high computational efficiency and reliability.

B. Personalized and Adaptive Learning

Personalized and dynamic learning methods utilized AI to analyze learners' knowledge, ability, and learning process to provide customized learning content, dynamic assessments, and personalized learning pathways. These methods effectively improve learner participation, learning efficacy, and academic excellence. The following section presents a survey of personalized and responsive learning works. Pradeesh et al. (2025) present an AI-supported personalized learning method by employing a knowledge-centric concept assessment. The subsystem incorporates retrieval-augmented generation, exercise-aware knowledge tracing, and Generative AI to analyze students' knowledge deficiencies, generate responsive evaluations, and provide personalized recommendations. Therefore, this research enhances learning outputs using concept-level assessments. Ma et al. [8] introduce an AI-based adaptive language learning system that consolidates natural language processing and a recurrent neural network to generate personalized learning approaches. The component analyses students' linguistic proficiency, speech patterns, and grammar to provide dynamic learning content, deliver real-time feedback, and recommend adaptive learning pathways. Thus, this research enhances student participation and English proficiency. Lou et al. [9] present an adaptive educational path-planning methodology that implements reinforcement learning and a cognitive graph to produce personalized learning paths. The subsystem continuously monitors learners' cognitive states, evaluates learning performance, and enhances content sequencing using predictive analytics. Thus, the presented system improves learning efficiency and educational outputs.

These approaches realize personalization based on knowledge tracing, learner proficiency analysis, and reinforcement-optimized path selection. Nevertheless, most of the proposed approaches concentrate on content adaptation and path personalization, while trustworthiness, explainability, and computational efficiency are insufficiently considered, which emphasizes the necessity of the developed approach, which is based on learner knowledge trace personalization.

C. Multi-Agent Systems in Education

Multi-Agent Systems (MAS) enable academic applications by allocating learning, tutoring, recommending, and assessment functions among several agents. Viswanathan et al. (2022) introduce a multi-agent learning system that coordinates multiple agents of personalized educational activities. Chang et al. [11] suggested a multi-agent approach to personalize education by dividing educational tasks among several agents. Rida et al. (2025) attempted to perform a multi-agent approach to achieving personalized educational services by assigning educational tasks to cooperating agents. Overall, the authors demonstrated that multi-agent approaches could be applied to perform specialized tasks, specifically in education. However, the majority of the works above employed a multi-agent approach without jointly considering trust evaluation, explainability, computational simplicity, and optimization, which limits adaptive and personalized learning by requiring extra agents to perform learning tasks, thus imposing

substantial overhead. As such, there is a need to develop a lightweight multi-agent architecture that targets accomplishing educational tasks while incorporating trust evaluation and explainability to ensure high-quality personalized learning.

D. Trustworthy and Explainable AI in Education

Trustworthy and explainable AI is necessary in education to confirm that AI-enabled decisions are transparent, responsible, explainable, and dependable for students and teachers. Latest studies focus on developing trustworthy governance, interpretable academic performance prediction, and reliable knowledge tracking strategies. The following is a survey of trustworthy and explainable AI in education. Evangelista et al. [13] introduce an innovative approach to a policy-as-code system that ensures trustworthy AI management in higher education. The component tracks institutional requirements and converts them into computable policies and uses a gatekeeper to evaluate system performance, calibration, fairness, explainability, and system limits. It additionally integrates fairness mitigation, SHapley Additive exPlanations (SHAP)-based interpretability, uncertainty-aware decisions, and tamper-evident audit procedures. Hence, the system allows explainable, accountable, and implementable AI governance during the development and deployment process. Raji et al. [14] introduce an explainable Machine Learning (ML) method to forecast the Deaf and Hard of Hearing learners' performance. The study deployed ML methods with Local Interpretable Model-agnostic Explanations (LIME) and SHAP to explain the forecasting results and detect important academic performance features. The combination of eXtreme Gradient Boosting (XGBoost), ExtraTrees, and Random Forest models resulted in 92.99% accuracy. Lu et al. (2024) proposed a trustworthy knowledge tracking model for an AI-enabled tutor system. Then, proposed three-level solutions to improve transparency, accountability, and interpretability through the integration of AI explanation techniques with educational systems. In turn, technical explanations are transformed into natural learning outcomes and presented through the ITS interface. Experimental evaluation shows that the presented model enhanced students' and teachers' trust and knowledge of the system.

These works relate to trustworthiness at various levels, from governance to prediction explainability and knowledge tracking. Yet, the emphasis is on providing an appropriate explanation or validation of each AI-based decision. These studies do not consider the combined evaluation of trust in the system with optimization of individual decisions and coordinated adaptation of agents. Consequently, it appears that the development of an integrated approach for a trustworthy, explainable, and learner-based system is needed.

E. Lightweight AI for Educational Applications

Lightweight AI models have emerged as capable approaches for educational applications, which enable real-time learner assessment while requiring lower computational resources. The following is a survey of lightweight AI for educational applications. Radharani et al. [16] present a light attention-based framework for detecting student attentiveness based on their facial expressions. MobileNet, together with channel and spatial attention, is deployed to identify facial features such as eye closure, mouth movements, and gaze. It achieves 91.2% accuracy with fewer parameters, which supports real-time classroom deployment [17]. Yu et al. (2025) present an IoT and edge-assisted system for computer-assisted learning. Here, IoT sensors are employed to record the engagement level of the students, while the edge devices process the data locally to provide immediate feedback with the aid of a lightweight stacked Long Short-Term Memory (LSTM) model. The system provides students with real-time feedback with 99.3% accuracy. Qorich et al. [18] introduce a context-based AI-enabled question generation approach for academic assessment that employs a lightweight transformer model. The subsystem combines a Convolutional Neural Network (CNN) module with a Bidirectional and Auto-Regressive Transformer (BART) base to enhance context-aware representation while preserving lower computational complexity; hence, this framework is suitable for a real-time educational environment.

These studies reduce computational costs by utilizing edge processing and lightweight transformers while ensuring effective educational outcomes. However, the efficiency is demonstrated in individual components, while the system-level factors, including agent execution, optimization, and trustworthiness, are limited [19]. Thus, more attention should be dedicated to developing educational frameworks that guarantee efficient model and end-to-end systems.

F. Metaheuristic Optimization for AI-Based Education

Optimization strategies employed in AI-powered learning include the use of heuristic search and evolutionary algorithms in optimizing learning pathways, content sequencing, learner modelling, and academic decision support. Most of the existing works used optimization algorithms like Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Library and Readers Algorithm (LRA) [22]. These algorithms enhance optimize learning approaches, educational assessment, quality of recommendations, and adaptive educational processes by identifying optimal solutions under multiple learning objectives. Subsequently, they enhance personalization, learner participation, and overall educational performance. However, most previous optimization methods mainly focus on optimizing individual educational tasks while needing iterative optimization and parameter tuning, which improves computational overhead and execution time. Furthermore, they provide limited support for lightweight multi-agent coordination, trust-based education content assessment, interpretable AI recommendations, dynamic prompt optimization, and computationally effective personalized learning [23]. The abovementioned limitations emphasize the requirement for an integrated optimization system capable of balancing personalization, trustworthiness, transparency, and computational efficiency in AI-driven educational technologies.

G. Research Gap and Positioning of the Proposed Framework

The literature review demonstrates that previous AI-enabled personalized learning systems have intelligent learner profiling, advanced tutoring, content recommendation, and

learning path generation. However, they do not adequately address high computational overhead, ineffective trust evaluation, static multi-agent coordination, limited explainable recommendations, and optimization methods that mainly focus on individual educational tasks instead of an integrated learning system [24]. Furthermore, recent methods offer limited optimization of AI agent selection, learning path adaptation, prompt generation, trust analysis, and computational resource allocation simultaneously, which reduces their practical deployment and scalability in resource-limited educational environments.

To address these challenges, the presented LITE-Trust system designs a lightweight trust-aware explainable multi-agent generative AI architecture optimized using the presented ELRA algorithm. In contrast to existing approaches, ELRA automatically coordinates educational agents, executes trust-based optimization, adapts personalized learning pathways, optimizes prompt selection, and minimizes computational complexity while integrating explainability and academic integrity assessment. Subsequently, the presented system positions itself as a unified, lightweight, trustworthy, and computationally effective approach for advanced AI-assisted personalized learning.

For the purpose of comparison, the capabilities in Table II are classified according to their explicit implementation in the reviewed studies. A ✓ indicates that the capability is directly implemented as a core component, ✗ indicates that it is not obviously addressed, and Partial indicates that the capability is considered indirectly or to a limited extent. The classification is based on the methods and objectives reported in the corresponding studies instead of inferred from their application context. Table II indicates that most of the studied works address some specific capabilities rather than the combination of their implementation. Personalization and optimization are the most commonly performed capabilities. Some papers address the coordination of multi-agent systems, but it is not fully explored, and only a few works mention multi-agent coordination, trust evaluation, XAI, and lightweight processing. In contrast, the LITE-Trust system combines several capabilities, which are partially addressed in the existing solutions, namely KSE-based personalization, ELRA optimization, dynamic multi-agent coordination, trust evaluation, XAI, and lightweight. However, it should not be assumed that other systems do not possess the mentioned capabilities.

TABLE II. RESEARCH GAP AND POSITIONING OF LITE-TRUST

Approach	Personalized	Multi-Agent	Trust	XAI	Lightweight	Optimization
Alyoussef et al. (2025)	✓	✗	Partial	✗	✗	✗
Algerafi et al. (2023)	✓	✗	✗	✗	✗	✗
Oh et al. (2025)	✓	✗	✗	✗	✗	✗
Pradeesh et al. (2025)	✓	✗	Partial	✗	✗	✗
Ma et al. (2025)	✓	✗	✗	✗	✗	✗
Lou et al. (2025)	✓	✗	✗	✗	✗	✓
Viswanathan et al. (2022)	✓	✓	✗	✗	✗	✗
Chang et al. (2026)	✓	✓	✗	✗	✗	✗
Rida et al. (2025)	✓	✓	✗	✗	✗	✗
Evangelista et al. (2026)	✗	✗	✓	✓	✗	Partial
Raji et al. (2024)	✗	✗	Partial	✓	✗	✗
Radharani et al. (2026)	✗	✗	✗	✗	✓	✓
Yu et al. (2025)	✓	✗	✗	✗	✓	✓
Kalwar et al. (2025)	✓	✗	✗	✗	✗	✓
Imamah et al. (2024)	✓	✗	✗	✗	✗	✓
[21] Zeng et al. (2022)	✓	✗	✗	✗	✗	✓
Proposed LITE-Trust	✓	✓	✓	✓	✓	✓

III. PROPOSED LITE-TRUST FRAMEWORK

A. Overall Architecture

The presented LITE-Trust framework initially collects learner information, including demographic characteristics, academic history, prior knowledge, learning preferences, assessment performance, engagement patterns, and interaction logs, to construct a dynamic learner profile as shown in Fig. 1.

The KSE subsystem then analyzes the obtained profile to estimate learners' competency levels and identify learning gaps. These outputs are then processed by the proposed ELRA, which identifies candidate educational policies, evaluates trust-aware recommendations, and optimizes educational decisions through personalized enhancement. Based on the optimized strategy, the lightweight multi-agent collaboration module adaptively activates and collaborates with the most suitable educational agents, which include the tutor, assessment,

feedback, recommendation, and academic integrity agents, based on the current learning context. Then, the explainability component analyzes the produced recommendations to provide transparent educational justifications, while the academic integrity module analyzes responsible AI usage, originality, critical thinking, and authentic learning behaviour. Finally, personalized learning recommendations are offered to learners, and continuous learner interactions are used to update learner profiles and knowledge state estimation, which enables adaptive, trustworthy, explainable, and computationally effective AI-enabled personalized learning.

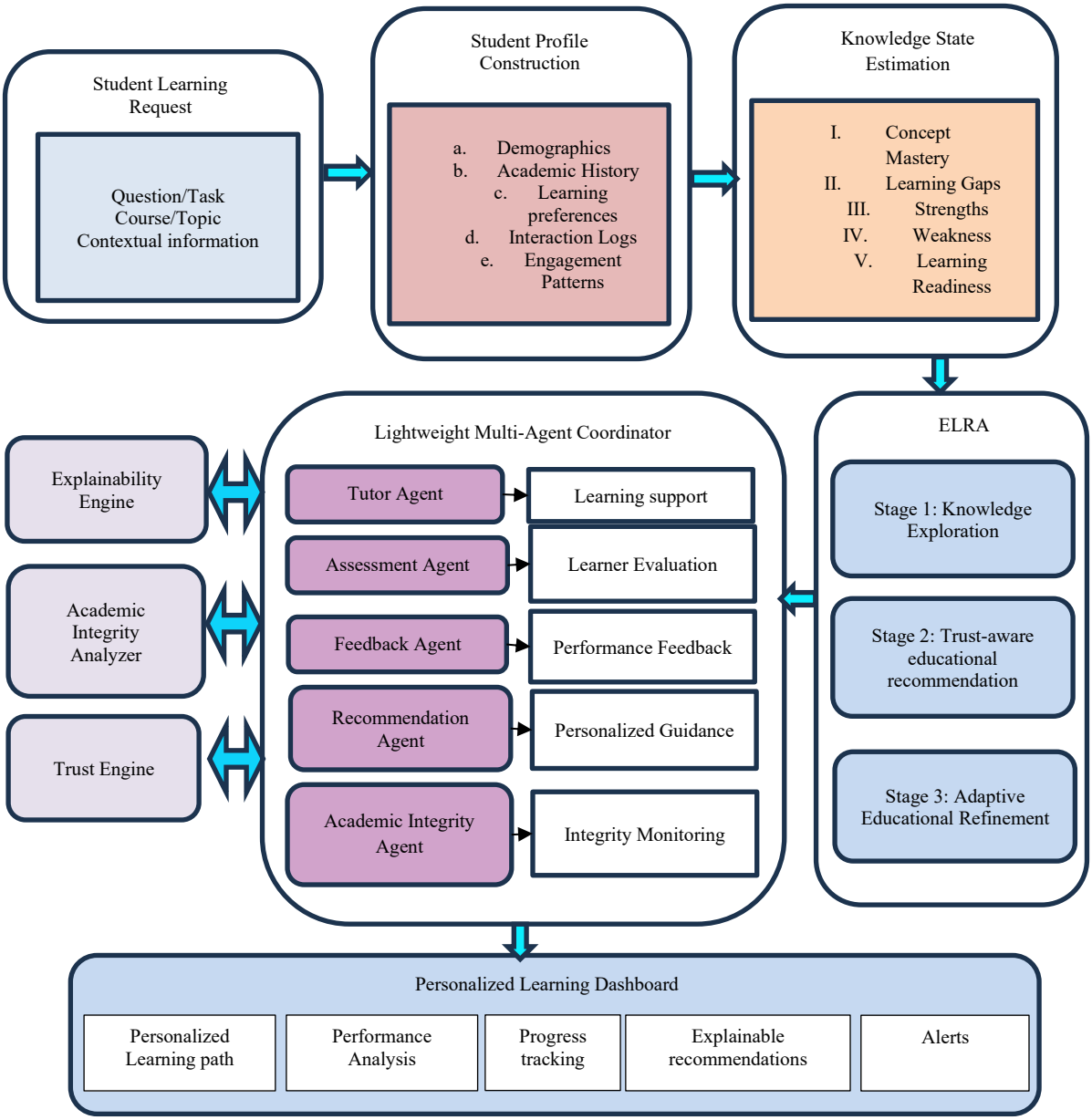


Fig. 1. Overall architecture of the LITE-Trust framework.

B. Student Profile Construction Modules

The student profile construction subsystem establishes a dynamic and comprehensive learner profile, supporting personalized educational decision-making within the LITE-Trust system. Previous AI-enabled learning systems often depend on static learner profiles or limited educational indicators, which makes it difficult to extract learners' developing knowledge, engagement, behavioural patterns, and individual learning preferences, which frequently leads to ineffective personalisation. To resolve the existing shortcomings, the proposed module combines heterogeneous learner data, which includes demographic characteristics (D_i), prior knowledge (P_{ki}), academic history (A_{Hi}), learning preferences (L_{Pri}), engagement patterns (E_{Pi}), assessment performance (A_{Pi}), and interaction logs (I_{Li}), into a unified learner profile that is continuously updated throughout the learning process. The learner profile is defined as,

$$LP_i = D_i, P_{ki}, A_{Hi}, L_{Pri}, E_{Pi}, A_{Pi}, I_{Li} \quad (1)$$

where, LP_i represents the learner profile. This representation consolidates multiple learner competencies into a unified learner profile, which provides a more holistic representation of learner competency than strategies that depend on limited academic indicators. The profile is continuously updated to reflect changes in the learner's progress, which is denoted as,

$$LP_i^{(t+1)} = LP_i^{(t)} \oplus L'_i \quad (2)$$

where, $LP_i^{(t)}$ represents the current learner profile, L'_i represents the newly produced learning evidence, and $\oplus$ represents the profile update operation. The newly obtained learning evidence consists of the assessment outcomes, engagement activities, and interaction records. Mathematically, the given formulation enables the learner profile to evolve by integrating newly generated learning evidence, thereby effectively capturing changes in learner competency, behaviour, and educational requirements throughout the learning process. Unlike static learner representations that mainly serve as descriptive inputs, the updated profile is then used as a decision-based state for KSE, which supports the estimated learner state to influence downstream personalized educational decisions. By combining heterogeneous learner information with a dynamic profile updating strategy, the presented subsystem overcomes the challenges of static learner modeling and restricted academic indicators.

C. Knowledge State Estimation Module

Following the student profile construction component, the KSE module is initiated to deliver an accurate and continuously updated representation of learner competency, which forms the basis for personalized learning decisions within the LITE-Trust framework. Previous AI-enabled learning systems mainly evaluate the learner's knowledge using static valuation scores or inadequate academic indicators, which makes them less efficient in extracting real-time learning development, detecting hidden learning gaps and accurately evaluating learner readiness for following concepts. To address these issues, the presented subsystem introduces a lightweight KSE model that employs the dynamically constructed learner profile together with multiple learner behaviour indicators, such as quiz performance, response time, learning activity patterns, revision frequency, and interaction history, to deliver a more reliable assessment of learner competency while preserving computational efficiency. The learner's knowledge state is denoted as a concept master vector, which is defined as,

$$KS_i = [ks_{i1}, ks_{i2}, \dots, ks_{in}], \;\; 0 \le ks_{ij} \le 1 \quad (3)$$

where, KS_i represents the knowledge state of the learner i, and ks_{ij} denotes the mastery level of the concept j, and n represents the number of learning concepts. This representation models learner competency at the concept level, which allows accurate detection of mastered and weak concepts instead of depending on a single assessment score. The mastery level of each concept is defined as,

$$w_{QP} + w_{RF} + w_{\mathrm{RT}} + w_{\mathrm{IH}} = 1 \quad (4)$$

Subject to

$$w_{QP} + w_{RF} + w_{\mathrm{RT}} + w_{\mathrm{IH}} \ge 0 \quad (5)$$

where QP, RF, RT, and IH represent the quiz performance, revision frequency, response time, and interaction history, respectively, while w_1, w_2, w_3, w_4 represent their respective weights. To ensure a consistent combination of heterogeneous learner indicators, the quiz performance (QP), revision frequency (RF), response time (RT), and interaction history (IH) are first normalized to a common range of $[0,1]$. Quiz performance is directly represented as a normalized score, whereas revision frequency and interaction history are scaled using min-max normalization. A lower response time generally indicates more efficient learner interaction; response time is normalized using an inverse min-max transformation. The normalized indicators are therefore defined as,

$$QP_{\{i,c\}}^{\{*\}} = QP_{\{i,c\}} \quad (6)$$

$$RF_{\{i,c\}}^{\{*\}} = \frac{RF_{\{i,c\}} - RF_{min}}{RF_{max} - RF_{min}} \quad (7)$$

$$RT_{\{i,c\}}^{\{*\}} = 1 - \frac{RF_{\{i,c\}} - RF_{min}}{RF_{max} - RF_{min}} \quad (8)$$

$$IH_{\{i,c\}}^{\{*\}} = \frac{IH_{\{i,c\}} - IH_{min}}{IH_{max} - IH_{min}} \quad (9)$$

Here, $QP^{\{*\}}, RF^{\{*\}}, RT^{\{*\}}$ *and* $IH^{\{*\}}$ denote the normalized indicators. The inverse transformation for response time ensures that a shorter response time corresponds to a higher normalized behavioural score. To provide a reproducible weighting scheme, the KSE module assigns

$$w_{\{QP\}} = 0.40, w_{\{RF\}} = 0.20, w_{\{RT\}} = 0.15, w_{\{IH\}} = 0.25 \quad (10)$$

$$w_{\{QP\}} + w_{\{RF\}} + w_{\{RT\}} + w_{\{IH\}} = 1 \quad (11)$$

The concept-level mastery score is then calculated as,

$$M_{\{i,c\}} = w_{\{QP\}} QP_{\{i,c\}}^{\{*\}} + w_{\{RF\}RF_{\{i,c\}}^{\{*\}}} + w_{\{RT\}RT_{\{i,c\}}^{\{*\}}} + w_{\{IH\}IH_{\{i,c\}}^{\{*\}}} \quad (12)$$

Here, $M_{\{i,c\}}$ represents the estimated mastery of the learner for the concept. The larger weight assigned to quiz performance reflects its direct relationship with demonstrated academic competency, while revision frequency, response behaviour, and interaction history provide complementary behavioural evidence. The resulting normalized and weighted score is subsequently used to construct the learner knowledge-state vector and readiness estimate. The weighted formulation incorporates both behavioural and academic learning evidence, which provides a more accurate evaluation of concept mastery than conventional score-based approaches. By incorporating the updated learner model that contains new evidence, the evaluated knowledge state can reflect changes in learner competency over time. The learner readiness score is calculated as,

$$LR_i = \frac{1}{n} \sum_{j=1}^{n} ks_{ij} \quad (13)$$

where, LR_i represents the learner readiness score, ks_{ij} represents the mastery level of concept j, and n represents the number of concepts. The highest value of the readiness score

demonstrates stronger prerequisite knowledge and greater readiness for following concepts. Unlike learner models that demonstrate knowledge using a single aggregate performance value, the presented KSE generates a concept-level state that can differentiate individual mastery differences over concepts and thus provide more specific information for following educational decisions. Furthermore, a knowledge graph captures conceptual relationships and concept dependencies, which enables the system to detect knowledge gaps in prerequisite concepts and evaluate learner readiness using structured knowledge reasoning. By integrating behavioural learning analytics from the learner profile with a computationally efficient architecture for knowledge graph-based reasoning. The proposed subsystem provides a continuously updated and trustworthy representation of learner knowledge. The derived concept mastery vector, learner readiness score, and identified learning gaps are then provided to ELRA as learner-state variables for educational refinement. These outputs define the personalized search space and support the optimization of academic decision variables such as AI agent selection, prompt template, learning sequence, difficulty of assessment, feedback depth, and computation resource allocation. This enables more accurate personalization and educational decision-making while preserving computational efficiency.

D. Educational Library and Readers Optimization (ELRA)

Following the KSE subsystem, the assessed learner knowledge state derived from the continuously updated learner profile is transferred to the ELRA subsystem for personalized educational decision-making. The KSE is used to estimate the students' current knowledge level and learning needs. ELRA detects the most appropriate combination of AI agents, prompts, learning sequence, valuation difficulty, feedback quality, and computational resources. Thus, KSE estimates the learners' current requirements, whereas ELRA establishes the main optimization strategy that determines how those requirements should be addressed through an appropriate educational configuration. Table III explains the purpose and novelty of the ELRA module, and Fig. 2 represents the flowchart of the ELRA subsystem.

ELRA improves the LRA, which captures information-seeking behaviour through exploration, peer-based guidance, and exploitation. However, the traditional LRA is a versatile optimizer and does not obviously integrate learner knowledge state, educational trust, and explainability. ELRA addresses these issues by introducing learner-state-guided exploration, trust-aware selection, dynamic exploitation, learner-specific configuration, and multi-objective optimization while retaining the basic LRA search principle.

1) Motivation: The selection of ELRA is guided by the precise requirements of personalized academic optimization. Traditional metaheuristic algorithms, such as GA, PSO, and other population-based methods, are primarily designed for general-purpose numerical optimization and do not naturally represent learner profiles, educational trust, or learner-oriented objectives. Alternatively, LRA provides a lightweight exploration-exploitation strategy motivated by information-searching behaviour, making it an appropriate basis for educational optimization. However, LRA does not obviously integrate learner state data or educational consistency. ELRA thus changes LRA to educational optimization by conditioning the search on the KSE-guided learner state, integrating trust into candidate evaluation and acceptance, adapting exploration to learner requirements, and collectively optimizing learning accuracy, explainability, trust, latency, and computational efficacy. This empowers ELRA to identify personalized and trustworthy educational configurations instead of generic optimization solutions.

2) Educational representation and initialization: The learner knowledge state gained from KSE is integrated into each academic configuration as,

$$I_i = [LP_i, KS, \mathbb{A}_i, \mathbb{P}_i, \mathbb{L}_i, \mathbb{D}_i, \mathbb{R}_i, \mathbb{F}_i] \quad (14)$$

where, LP_i represents the dynamic learner profile, KS represents the assessed learner knowledge state, while $\mathbb{A}_i, \mathbb{P}_i, \mathbb{L}_i, \mathbb{D}_i, \mathbb{R}_i, \mathbb{F}_i$ represents the AI agent, template of prompt, learning sequence, difficulty of assessment, computational resources, and feedback depth, respectively. These variables constitute the decision dimensions optimized by ELRA. In contrast to traditional optimizers, this representation explicitly relates the optimization variables to the learner's estimated state, which enables ELRA to search for personalized instead of standard educational configurations. This learner-state conditioning creates the search spaces as an education-specific decision space, in which optimal configurations are estimated based on the requirements of the current learner instead of general-purpose optimization solutions. The initial candidate population is defined as,

$$I_i^{(0)} = lb + r_i(ub - lb) \quad (15)$$

where, ub and lb represent upper and lower bounds and $r_i \sim u(0,1)$. This allocates candidate configurations over the search space and maintains diversity during initialization.

3) KSE-guided knowledge exploration: To adapt the exploration to educational requirements, ELRA changes the peer-guided LRA search using the learner state and trust data,

$$I_i^{P1} = I_i + \alpha_t(1 - 2r)\big(I_j - I_i\big) + \delta KS + \gamma\big(\mathcal{T}_j - \mathcal{T}_i\big) \quad (16)$$

where, I_i^{P1} represents the exploratory candidate configuration generated for the candidate i, I_i represents the current candidate configuration, I_j is a peer candidate reader, $r \sim u(0,1)$, $\mathcal{T}_j, \mathcal{T}_i$ represents the trust scores of the current and peer configurations, α_t regulates the exploration magnitude, δ regulates the learner knowledge state influence, and γ denotes the trust-based exploration coefficient, which controls trust-based movement.

The conventional LRA update, which primarily uses peer-candidate information to determine the search direction, is augmented by the proposed ELRA update introduces two additional sources of optimization guidance: the learner knowledge state and the trust difference between candidate configurations. The learner-state term biases exploration toward configurations that address the learner's identified knowledge gaps and learning requirements, whereas the trust

term discourages movement toward configurations with lower educational reliability. The search direction is no longer determined solely by the geometric relationship between candidate solutions. Instead, ELRA produces a learner- and trust-conditioned search trajectory. This modification is algorithmic because it changes the candidate-generation rule itself and directly affects which configurations are explored during optimization.

Unlike the traditional LRA update, the presented formulation directs exploration based on both learners' needs and trust differences, instead of relying solely on peer information. This change prevents the exploration process from treating all candidate configurations equally and enables the search direction to remain stable with both the learners' estimated needs and the trustworthiness of the educational configuration.

4) Trust-aware evaluation: Every explored configuration is estimated using the trust value of the candidate configuration, which is defined as,

$$\mathcal{T}_i = w_1 FC_i + w_2 + w_3 C_i + w_4(1 - HP_i) + w_5 EQ_i + w_6 EC_i \quad (17)$$

where, $FC_i, CA_i, C_i, HP_i, EQ_i, and\ EC_i$ represents factual correctness, curriculum alignment, confidence, hallucination probability, explanation quality, and educational completeness, respectively, with $\sum_{i=1}^{6} w_i = 1$. A candidate undergoes refinement only when $\mathcal{T}_i \geq \mathcal{T}_{min}$. This trust limitation introduces a quality gate into the optimization process, which prevents candidates with unreliable or low-quality educational reliability from proceeding to refinement, instead of enabling fitness enhancement alone to determine candidate progression.

5) Trust and state-guided adaptive exploitation: For trusted configurations, ELRA performs adaptive optimization as

$$I_i^{P2} = I_i + \varepsilon_t(1 - 2r)\frac{\|ub - lb\|}{dt} + \omega\left(\mathcal{T}_j - \mathcal{T}_i\right) + \gamma(KS - EC_i) \quad (18)$$

where, I_i^{P2} represents the optimized candidate configuration, I_i represents the current candidate, ε_t regulates the exploitation step, d represents the decision variables count, t denotes the iteration number, r denotes a random variable; it is retained in the equation, ub, lb denotes the upper and lower bounds, γ denotes the learner-state influence coefficient, and EC_i represents the knowledge state needed to be associated with the current configuration. The gradually decreasing step size allows broader exploration in early iterations and fine-tuning in later iterations. The trust and state terms limit the optimization direction based on configuration reliability and learner-specific needs. This, while retaining the regular exploitation principle, is retained from LRA; ELRA conditions the optimization process on the evaluated knowledge state and configuration trustworthiness instead of depending only on numerical fitness improvement.

6) Trust-aware acceptance and convergence: A candidate substitutes the current configuration only when it meets trust, fitness, and learner-specific conditions,

$$I_i(t+1) = \begin{cases} I_i^{P2}, & \mathcal{T}_i^{P2} \geq \mathcal{T}_{min} \wedge F_i^{P2} < F_i \wedge \mathcal{A}(KS, I_i^{P2}) > \mathcal{A}(KS, I_i) \\ I_i, & otherwise \end{cases} \quad (19)$$

where, $I_i(t+1)$ represents the updated candidate configuration, I_i^{P2} denotes the optimized candidate $\mathcal{T}_i^{P2}$ represents the optimized candidate trust score, $\mathcal{T}_{min}$ denotes the minimum acceptable trust threshold, F_i^{P2} represents the optimized candidate fitness. ELRA minimizes the fitness function $F_i^{P2} < F_i$ indicates an objective improvement. $\mathcal{A}(KS, I_i)$ calculates the alignment between the learner's knowledge state and academic configuration. ∧ represents logical AND, requiring all three conditions simultaneously. Therefore, ELRA accepts a promising solution only if it meets the trust requirements and improves the overall objective, and achieves better alignment with the learner state. This acceptance strategy prevents a candidate from being selected solely because of numerical objective enhancement, thus maintaining learner-specific suitability during the final solution-selection stage. The process ends when the highest iteration count is reached, or the enhancement becomes negligible. To ensure mathematical consistency and reproducibility, all learner-state, trust, alignment, and continuous configuration variables used by ELRA are represented in compatible normalized spaces. Let the learner knowledge state be represented by the concept mastery vector.

$$K_i = \left[k_{\{i1\}}, k_{\{i2\}}, \dots, k_{\{iC\}}\right]^{\{T\}} \in [0,1]^C \quad (20)$$

Here, C denotes the number of learning concepts and $kick_{\{ic\}}$ represents the estimated mastery of learner i for concept c. The corresponding learner-state vector used by ELRA is a C-dimensional normalized vector. An educational configuration is represented by

$$X_i = [a_i, p_i, l_i, d_i, r_i, f_i]^{\{T\}} \quad (21)$$

Here, a_i denotes the selected AI-agent configuration, p_i denotes the prompt-template configuration, l_i denotes the learning-sequence configuration, d_i denotes assessment difficulty, r_i denotes computational-resource allocation, and f_i denotes feedback depth. For numerical compatibility, continuous decision variables are normalized to [0,1] as

$$x_{\{ij\}}^{\{*\}} = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}} \quad (22)$$

Here, x_j^{max}, x_j^{min} denote the lower and upper bounds of the j-th decision variable. Categorical variables, such as agent and prompt selections, are represented by their predefined valid indices or one-hot configurations and are mapped back to their corresponding discrete values after optimization. The exploration and exploitation operations are performed in a common normalized decision space. The learner-state influence is incorporated through a dimension-compatible mapping

$$g(K_i, X_i) = W_{KK_i} - W_{XX_i} \quad (23)$$

Here, W_{KK_i}, W_{XX_i} are dimension-matching projection matrices that map the learner-state and configuration

representations into the same d-dimensional optimization space. The trust score is normalized to the interval [0,1] and is calculated as:

$$T(X_i) = w_F F_i + w_C C_i + w_Q Q_i + w_E E_i + w_H(1 - H_i) + w_G G_i,\ \sum_m w_m \geq 0 \quad (24)$$

Here, $F_i, C_i, Q_i, E_i, H_i\ and\ G_i$ represent factual correctness, curriculum alignment, confidence, explanation quality, hallucination probability, and educational completeness, respectively. The transformation $(1 - H_i)$ converts hallucination probability into a positive reliability measure. The learner-configuration alignment is similarly normalized as

$$A(K_i, X_i) = 1 - \frac{\|g(K_i, X_i)\|_2}{\sqrt{d}} \quad (25)$$

A larger value of A indicates stronger compatibility between the learner's estimated knowledge requirements and the candidate educational configuration. Using these normalized quantities, the ELRA fitness function is expressed as

$$J(X_i) = w_L L(X_i) + w_T T(X_i) + w_A A(K_i, X_i) - w_R R(X_i) - w_C C(X_i) \quad (26)$$

Here, L denotes learning-performance utility, T denotes trust, A denotes learner-configuration alignment, R denotes response latency, and C denotes computational cost. All terms are normalized to [0,1], and the weights satisfy

$$w_L + w_T + w_A + w_R + w_C = 1 \quad (27)$$

The ELRA optimization with best configuration is consequently defined as

$$X^* = \underset{X \in \Omega}{\operatorname{argmax}} J(X) \quad (28)$$

This formulation establishes a common normalized representation for learner state, educational configuration, trust, and alignment and explicitly defines the feasible search space, objective function, acceptance conditions, and convergence criterion.

7) Multi-objective fitness function: ELRA collectively optimizes learning accuracy, trust, explainability, latency, and computational cost.

$$F(I) = \aleph_1(1 - L_A) + \aleph_2(1 - \mathcal{T}) + \aleph_3(1 - E) + \aleph_4 L + \aleph_5 C_C \quad (29)$$

where, $L_A, \mathcal{T}, E, L, and\ C_C$ represents normalized learning accuracy, normalized trust, normalized explainability, normalized latency, and normalized computational cost, respectively, and $\aleph_1, .., \aleph_5$ denotes the objective weights, $\sum \aleph_i = 1$. In order to compare different objectives with each other, all five objectives were normalized to a scale of [0,1]. The benefit objectives, including learning performance, trust, and explainability, were normalized using min-max normalization as $X_i' = (X_i - X_i^{min})/(X_i^{max} - X_i^{min})$. The cost objectives, including latency and computation cost, were inversely normalized as $X_i' = (X_i^{max} - X_i)/(X_i^{max} - X_i^{min})$.

In addition to modifying candidate exploration, ELRA changes the candidate evaluation and acceptance process. For a candidate configuration XX, the optimization objective considers learning performance, trustworthiness, explainability, latency, and computational efficiency. This constrained acceptance mechanism distinguishes ELRA from conventional LRA, in which candidate evolution is primarily governed by the search and fitness mechanism. In ELRA, an improvement in the numerical objective alone is insufficient when the candidate does not satisfy the required trust and learner-alignment conditions. Thus, higher normalized values indicate better performance for all objectives, while the original cost measures are minimized. Afterwards, the normalized objectives were aggregated using the weighted objective configurations in Table XX, where the weighting groups represent different priorities among learning performance, trust, and efficiency. This design supports the sensitivity analysis to examine how changes in objective priorities affect the overall fitness and learning performance. The optimal configuration is obtained as,

$$I' = \arg\min_I F(I) \quad (30)$$

Subject to

$$\mathcal{T}(I) \geq \mathcal{T}_{min} \quad (31)$$

where, $\mathcal{T}_{min} = 0.70$. This mathematical representation balances personalized learning efficiency, educational consistency, interpretability, response effectiveness, and resource consumption, thus treating educational optimization as a multi-criteria decision problem instead of relying solely on learning performance. In contrast to the conventional LRA objective, which is not formulated around these academic criteria, ELRA assesses promising configurations based on a unified educational objective that collectively includes effectiveness, interpretability, trustworthiness, computational efficiency and responsiveness.

The computational complexity is approximately $O[tn(F_C + T_C)]$, where t denotes the iteration count, n represents population size, F_C represents fitness cost, and T_C represents trust cost. The additional trust and learner-state evaluations introduce minimal computational cost while maintaining the lightweight population-based structure of LRA. Hence, ELRA is not only a relabeling of LRA; its exploration, optimization, and acceptance strategy explicitly integrate learner-state and trust information to generate personalized education configurations. The conventional LRA performs exploration and exploitation primarily through peer-guided candidate movement within a general optimization space. Its search behavior is determined by the current candidate, a peer candidate, and the corresponding search coefficients. In contrast, ELRA retains the basic exploration–exploitation principle of LRA but introduces learner-state conditioning, trust-aware candidate evaluation, and learner-specific acceptance constraints. Therefore, ELRA modifies the search dynamics rather than merely applying LRA to an educational task. ELRA extends this formulation by conditioning the candidate movement on the learner knowledge state and the trust difference between configurations. The search direction in ELRA is jointly determined by peer information, learner requirements, and trustworthiness. ELRA introduces a trust-

gated refinement mechanism. A candidate configuration is permitted to enter the subsequent optimization stage only when

$$T(X_{it}) \geq \tau_T, T(X_i^t) \geq \tau_T \quad (32)$$

Here, τ_T is the predefined trust threshold. The accepted candidate is then evaluated according to both optimization quality and learner-configuration alignment. This introduces a constraint that is absent from the conventional LRA search process. The principal algorithmic distinction of ELRA is the transformation of an unconstrained peer-guided search into a learner-conditioned, trust-gated, and multi-objective educational optimization process.

TABLE III. PURPOSE AND NOVELTY OF ELRA MODULE

ELRA Module	Conventional LRA	Proposed ELRA Extension	Technical implication
Educational reader	General candidate/search representation	It represents AI agent, prompt, learning, feedback, assessment, and resources	Generates a learner-specific academic-search space
Knowledge exploration	Peer-guided exploration based on information-seeking patterns	Exploration is driven by the KSE-guided learner knowledge state and trust in information	Guides the search toward configurations aligned with the learners' current requirements
Trust evaluation	Candidate quality is mainly established through optimization fitness	Evaluates factual correctness, hallucination, curriculum alignment, explanation quality, and educational completeness	Presents educational trustworthiness in candidate evaluation
Candidate selection	Candidate evolution follows the traditional search and fitness strategies	Candidates should satisfy a minimum trust threshold before further optimization	Prevents untrustworthy educational configurations from entering additional optimization
Adaptive exploitation	Traditional exploitation of promising search regions	Exploitation is dynamically guided by learner state, trust, and iteration-dependent step size	Guides fine-tuning toward learner-aligned and reliable configuration while gradually reducing the search step
Candidate acceptance	Selection is mainly based on enhancement within the optimization process	Acceptance includes trust, objective enhancement, and learner-configuration alignment	Maintains both solution quality and learner-specific fitness
Objective formulation	General optimization objective	Collectively optimizes learning accuracy, explainability, trust, latency, and computational cost	Adapts the search into an academic-specific multi-objective optimization problem
Final solution	Generally optimized candidate	Trustworthy configuration aligned with the learner's knowledge state and academic requirements	Generates a learner-specific, resource-aware, and trustworthy academic configuration for personalized decision-making

Pseudocode of ELRA module

Input: learner profile $L_P, , lb, ub,$ knowledge state KS, population size n, maximum iteration $\mathcal{T}_{max}$, trust threshold $\mathcal{T}_{min}$, and convergence tolerance ε
Output: optimal educational configuration I'
Initialize population $I_i^{(0)}$ using Eq. (8) $i = 1, \dots, n$
Evaluate fitness $F(I_i)$ using Eq. (13) and trust score $\mathcal{T}_i$ using Eq. (10) for each I_i
Select the optimal trusted configuration I'
For t=1 to $\mathcal{T}_{max}$ do
For i=1 to n do
Select optimal configuration I_j, where $j \neq i$
Generate experimental configuration I_i^{P1} using Eq. (9)
Compute trust score $\mathcal{T}_i^{P1}$ using Eq. (10)
If $\mathcal{T}(I) \geq \mathcal{T}_{min}$ then
Generate optimized configuration I_i^{P2} using Eq. (11)
Compute fitness $F(I)$ using Eq. (13), trust score $and\ \mathcal{T}_i$ using Eq. (10), and learner-configuration alignment
If $\mathcal{T}(I) \geq \mathcal{T}_{min}$ the learner-state requirements satisfy Eq. (11), then
Update as the trusted candidate with minimum fitness
End if
End if
End for
Update the population toward the trusted candidate using Eq. (9)
If the convergence condition is satisfied, then break
End for
Return I'

E. Lightweight Multi-Agent Coordinator

Following the KSE and ELRA subsystems, the optimized academic configuration produced by ELRA, such as selected AI agent prompt templates, learning sequence, assessment difficulty, feedback depth, and computational resource allocation, is directly fed into the activation strategy of the lightweight multi-agent coordinator. The multi-agent coordinator is responsible for deciding which agents are responsible for delivering it, which includes the tutor, assessment, feedback, recommendation, and academic integrity agents. However, all agents being active for each interaction would involve high computational costs and delays; the presented module introduces context-adaptive agent activation. The tutor agent offers explanations and guidance, the assessment agent computes learner performance, the feedback agent provides feedback to the learner, the recommendation agent selects relevant learning resources, and the academic integrity agent validates content and activities for academic integrity issues. Their specialized roles enable educational tasks to be dynamically allocated based on the learners' current requirements. Every educational agent is implemented as a task-specific role, based on a shared Generative Pre-trained Transformer 4 omni (GPT-4o) mini language model, rather than a separate model. Every agent takes the learner knowledge state, ELRA-optimized configuration, and task-specific context as inputs, and outputs role-specific responses, such as explanations, evaluations, feedback, learning-resource recommendations, or integrity checks, respectively. The coordinator calculates the relevance of every agent from the

current learning context and learner-state/ELRA configuration, activates the agents whose relevance scores exceed the activation threshold, and respects the quality and computational-cost constraints (Eq. (16)-(18)). The relevance score is determined according to the compatibility between the current academic task, the KSE-derived learner knowledge state, and the ELRA-optimized academic configuration. Thus, agent activation is learner- and task-dependent instead of based on a derived execution sequence. The active agent set is defined as,

$$AG_t = \{a_j | r_j(KS, I', C_{Et}) \geq \theta_j\} \quad (33)$$

where, AG_t represents the set of active agents at iteration t, a_j represents the j-th educational agent, $r_j(.)$ represents the relevance score of agent j, C_{Et} represents the current learning context, I' denotes the ELRA-optimized educational configuration, and θ_j denotes the activation threshold. Thus, agents are activated only when their relevance scores exceed the specified threshold. This KSE-and ELRA-based selection minimizes unnecessary agent execution and identifies the gap in redundant multi-agent processing and maximizes response latency. The selected agent outputs are denoted as,

$$OP_t = Cf(\{OP_j : j \in AG_t\}, KS, I') \quad (34)$$

where, OP_t represents an active agent j output and $Cf(.)$ represents the agent coordination function. By integrating KS and I', the coordinator produces agent outputs based on the learners' state and personalized educational requirements, thus identifying isolated agent operation and unreliable responses. To reduce agent utilization while preserving educational quality, the optimal active set is defined as,

$$AG_t' = \arg\min_{AG_t} [\lambda_c | AG_t + \lambda_l L_t] \; s.t \; RQ(AG_t, KS, I') \geq RQ_{min} \quad (35)$$

where, $|AG_t|$ represents the active agent count, $\lambda_c, and\ \lambda_l$ represents the weight values of computational cost and response latency. The coefficients satisfy $\lambda_c, \lambda_l \geq 0, \lambda_c + \lambda_l = 1$. L_t denotes response latency, and RQ denotes educational response quality, and RQ_{min} denotes the minimum acceptable response quality.

This mathematical representation identifies the minimum set of agents that meets the required quality, which provides the major contribution of lightweight, quality-based multi-agent implementation. Thus, agent selection is formulated as a forced active -set decision that collectively considers educational quality and processing cost, instead of evaluating the relevance of each agent individually. Overall, the subsystem contributes learner-based, ELRA-driven, dynamically activated, and quality-based multi-agent coordination, which addresses redundant computation, latency, isolated agent operation, and unreliable educational responses. Table IV emphasizes the practical contribution of every specialized agent and its role in the designed framework. In contrast to traditional multi-agent systems that mainly combine individual agents, the presented coordinator assigns learner-personalized responsibilities and adaptively activates agents based on the KSE-based state and ELRA-optimized configurations.

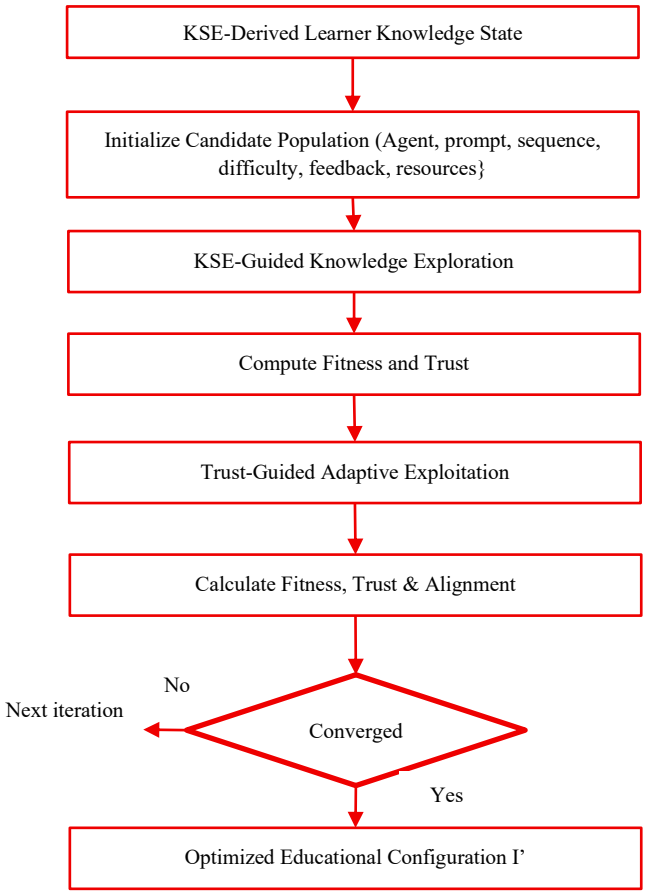


Fig. 2. Flowchart of ELRA module.

TABLE IV. CONTRIBUTION OF MULTI AGENTS

Agents	Proposed Contribution
Tutor Agent	Provides explanations and guided learning based on the KSE-based learner state, which enables personalized knowledge support.
Assessment Agent	Produces and evaluates assessments using the ELRA-adapted difficulty, which allows learner -adaptive assessment.
Feedback Agent	Delivers targeted corrective and formative feedback with an appropriate level of depth, which supports knowledge-gap-specific learning support.
Recommendation Agents	Chooses learning resources and activities based on KS, and I', which enables optimization-based resource personalization.
Academic Integrity Agent	Evaluates academic content and activities before delivery, which provides trust-aware integrity verification throughout the learning pipeline.

F. Explainability Engine

Following the Lightweight Multi-agent coordinator, the outputs of the selected agents OP_t along with the KSE-based learner state KS_i and ELRA-optimized configuration I' are offered to the explainability engine. While personalized AI decisions can enhance learning outputs, their limited transparency may reduce trust among learners and instructors. Thus, the presented engine produces explanations, reasoning, and confidence for each academic decision, which identifies the interpretability gap in AI-based personalized education. The explanation is expressed as,

$$EX_t = \varepsilon(KS_i, I', OP_t, E_{ct}) \tag{36}$$

where, EX_t represents the produced explanation, $\varepsilon(.)$ represents the explainability function, I' represents the ELRA-optimized configurations, and E_{ct} represents the learner's current educational context. The formulation is conditioned on the learner state, optimized configurations, and agent outputs, explaining why a concept was learned, why assessment difficulty was chosen, why a revision action was taken, or why a learning resource was recommended, thus addressing the issue of context-free recommendations. The decision reasoning and confidence are obtained as,

$$DR_t = DR(KS_i, I', OP_t), S_t = S(DR_t) \tag{37}$$

where, DR_t represents decision reasoning and S_t represents the respective confidence score. $DR\,(.)$ represents the reasoning generation function, $S(.)$ represents the confidence estimation function; both quantities are derived from the learner state, optimized configuration, and coordinated agent output. Hence, the engine explains the rationale for each decision and its confidence level, thereby enhancing educational transparency and instructor confidence. The final explainable output is formulated as,

$$Z_t = \{EX_t, DR_t, S_t\} \tag{38}$$

where, Z_t collectively represents explanation, reasoning, and confidence. This formulation connects the generated explanation to the information that contributed to the educational decision, which facilitates the rationale to reflect the learner-specific state and optimal educational configuration instead of generating individual generic explanations. Hence, the presented engine provides learner-state-aware and ELRA-based explanations, reasoning, and confidence, thereby bridging the transparency and trust gap in personalized academic decision-making.

G. Trust Evaluation Engine

Following the explainability engine, the trust evaluation engine executes final trust validation of the educational response produced by the optimal agents. While ELRA already computes trust during optimization, that estimation is used to select optimal candidate configurations. Unlike the presented trust evaluation engine, which validates the actual produced response prior to delivery, it addresses the gap between trustworthy configuration selection and trustworthy final academic output. The final response trust score is expressed as,

$$T_{St} = w_{fc}FC_t + w_{ca}CA_t + w_h(1 - H_t) + w_cC_t \tag{39}$$

Subject to:

$$w_{fc} + w_{ca} + w_h + w_C = 1 \tag{40}$$

where, T_{St} represents the final response trust score of the generated response, FC_t represents factual correctness, CA_t represents curriculum alignment, H_t represents hallucination probability, and C_t denotes confidence. While w_{fc}, w_{ca}, w_h, w_C represents their respective weights. The term $1 - H_t$ transforms the estimated hallucination probability into a respective reliability measure. Hence, the mathematical representation collectively evaluates factual reliability, curriculum suitability, hallucination risk, and confidence, instead of depending solely on model-generated confidence. A response is accepted only when

$$T_{St} \geq T_{S_{min}} \tag{41}$$

where, T_{St} represents the final response trust score, and $T_{S_{min}}$ represents the minimum acceptable response threshold. All trust components are normalized to [0,1], making $T_{St} \in [0,1]$. A response satisfying $T_{St} \geq T_{S_{min}}$ is accepted for delivery, whereas a response below this threshold is rejected or refined before delivery. In the experimental configuration, the trust threshold is fixed at 0.70. For this final step, the produced response is evaluated against both the corresponding educational knowledge and reference information for the associated learning task. Factual correctness is established by comparing the derived responses to the provided reference information. Meanwhile, the degree of alignment with the curriculum is determined by the target concept and its requirements. Thus, hallucination probability is defined as the ratio of inaccurate information that was found in comparison with the reference information during fact-checking. The confidence score is extracted from the model confidence assessment. Finally, these metrics are used to inform the response-level acceptance decision. The threshold therefore functions as a response-level acceptance limit, which makes trust evaluation an active control strategy instead of only a post -hoc reliability indicator. Hence, the presented engine acts as a final trust-aware integrity control, which prevents undependable educational responses from reaching the learner. Unlike ELRA, in which trust drives the candidate configuration selection and optimization, this component analyses the final agent's response. This two-stage trust strategy boosts the developed framework by incorporating both learning processes: the trust-aware educational optimization and final fact-checking step, which improves the accuracy of hallucination-free and curriculum-aligned educational content while enabling more reliable personalized decision-making.

H. Academic Integrity Analyzer

Following the trust evaluation engine, the academic integrity analyzer estimates whether AI-assisted learning maintains originality, critical thinking, independent problem-solving, and authentic learning behaviour. Traditional integrity methods focus primarily on plagiarism, which may overlook excessive AI dependency and reduced learner independence. Hence, the introduced analyzer aims to mitigate these issues by monitoring the authenticity and adequacy of AI-assisted learning. The educational integrity score (I_{At}) is expressed as,

$$I_{At} = w_1 OR_t + w_2 CT_t + w_3 PS_t + w_4 AL_t \quad (42)$$

Subject to

$$w_1 + w_2 + w_3 + w_4 = 1 \quad (43)$$

where, OR_t represents originality, CT_t represents thinking, PS_t represents independent problem-solving, and AL_t represents authentic learning behaviour, each estimated from relevant learner interaction and performance patterns available in EdNet and OULAD. $w_1, \dots, w_4$ represents corresponding weight values $w_k \geq 0 \sum w_k = 1$. The weighted formulation integrates multiple behavioural indicators, thereby delivering a more holistic integrity assessment than plagiarism detection alone. AI dependency is evaluated as,

$$AD_t = \frac{\zeta_{AI}}{\zeta_{AI} + \zeta_I} \quad (44)$$

where, AD_t represents the estimated AI-dependency score, ζ_{AI} represents AI-assisted interactions, and ζ_I represents independent learning interactions. A maximum AD_t represents greater AI dependency. Higher AD_t indicates greater reliance on AI-assisted interactions. The integrity need is satisfied when $AI_t \geq AI_{min}$, $AD_t \leq AD_{max}$ where AI_{min} represents the minimum integrity score and AD_{max} represents the maximum acceptable AI dependency. These constraints ensure that AI assistance does not take over from genuine human learning. The indicators identified by the academic integrity analyzer should be regarded as behavioural estimates rather than measures of academic integrity. In fact, EdNet and OULAD provide information on the learning process and its outcomes, from which integrity and the extent to which students rely on AI are inferred. Put another way, since the input data cannot provide ground truth about academic integrity and AI reliance, the analyzer does not claim to directly assess these aspects, but rather it identifies learning behaviours that may indicate academic integrity issues and/or reliance on AI. Therefore, the contribution of the proposed work is an AI-aware, behaviour-based integrity check that builds upon the preceding trust verification and aims to bridge the plagiarism-centric gap in academic integrity assessment.

The academic-integrity component is currently evaluated using behavioral indicators derived from learner interaction patterns rather than independently annotated ground-truth integrity labels. Therefore, its outputs should be interpreted as risk indicators for potential academic-integrity concerns, rather than definitive evidence of misconduct. The framework does not claim to establish academic-integrity violations solely from these behavioral estimates. Validation against expert annotations or independently labeled behavioral indicators is identified as an important direction for future evaluation.

I. Personalized Learning Dashboard

Following the academic integrity analyzer, the personalized learning dashboard introduces the learners' learning analytics, personalized learning recommendations, concept-level progress, knowledge gap alerts, trust alerts, and academic-integrity alerts using a lightweight interface. Traditional educational dashboards provide learning information to a learner in a non-adaptive manner. Instead, the presented dashboard employs KSE-derived learner state, ELRA-optimal configuration, multi-agent outputs, trust assessment, and academic integrity results to provide an integrated learner view. The dashboard limits the information that is relevant to the learning context, such as the level of mastery, recommendations of resources, learning needs, assessment results, trust alerts, and academic-integrity alerts, thus enabling faster processing of the information and interaction.

The novelty of LITE-Trust lies in the explicit dependencies among the proposed methods rather than the individuality of each technique used. In particular, while the continuously updated learner profile informs KSE, KSE informs the learner's knowledge state for ELRA, the latter informing the educational configuration that determines the agent's selection. The response is then evaluated by the trust engine. As such, it can be argued that the mathematical formulations describe a learner-state-driven decision-making process where each step informs the next one rather than several independent procedures combined together.

The LITE-Trust framework is designed to perform lightweight execution through compact KSE, limited iterations of ELRA, selective agent activation, and optimized prompt and retrieval processes. First, KSE reduces subsequent operations by using a compact knowledge-state representation. ELRA further limits search by considering a fixed population size and iterations. Finally, dynamic coordination allows for activating only specific agents according to the incoming request. In addition, prompt optimization and dense-vector-retrieval reduce additional computational burdens. Hence, the efficiency gains mainly result from reduced representation, limited optimization step, selective execution, and retrieval-generation costs. The reported parameter count excludes the parameters of the external LLM employed by LITE-Trust.

J. Computational Complexity Analysis

The computational efficacy of LITE-Trust is achieved through the implementation of lightweight KSE representation, reduced ELRA search, optimal agent activation, and optimized prompt and retrieval processing. The overall computational complexity is then decomposed based on the major processing components of LITE-Trust. The KSE component requires $O(nl)$, where n represents the count of learner records, andan l represents the average learner-interaction sequence length. This follows because each of the n learner records processes an average of l interaction elements. ELRA requires $O(tnF_C)$, where t denotes the iteration count, n represents population size, F_C represents fitness estimation cost for one candidate solution. Therefore, n candidates are evaluated during each of t iterations. Dynamic agent selection ,requires $O(V + k)$ in the worst case and $O(k)$ after selective activation, where V is the total number of available agents and k is the number of activated agents. Since LITE-Trust achieves only the agents required for a particular learis ner request, $k \ll V$ in a typical execution, thereby avoiding unnecessary agent computation and reducing inference latency and utilization. Trust evaluation requires $O(kq)$, where q represents the cost of computing the trust-related indicators. Explainability processing requires $O(e)$, representing the cost of generating and processing the corresponding explanations, where e denotes the number of explanatory factors. Finally, retrieval and generation require

$(\log N + r + C_g)$, where N represents the indexed knowledge-base size, r represents the number of retrieval items, and C_g represents generation cost. Consequently, the overall computational complexity is derived as the sum of the component complexities $C_{LITE-Trust} = O(nl + tnF_C + V + k + kq + e + \log N + r + C_g)$. This component-wise formulation reflects the actual operations performed by each module. The presented lightweight design reduces computational complexity through compact KSE representation, bounded ELRA search, selective agent activation, and optimized retrieval and generation, thereby reducing unnecessary computation and inference latency while preserving the required educational quality.

To provide a more realistic efficiency assessment, the computational analysis considers the complete execution pipeline rather than only the local algorithmic complexity. The evaluation accounts for GPT-4o mini API invocation, embedding generation, retrieval operations, multiple-agent execution, trust verification, and ELRA optimization iterations. For each request, end-to-end latency is considered together with local processing time, API response time, network communication overhead, and the number of activated agents. External inference and API usage costs are also recognized separately from the computational cost of the locally deployed components. This end-to-end assessment provides a more representative measure of the practical efficiency of LITE-Trust.

IV. Results and Discussion

A. Experimental Environment and Datasets

The LITE-Trust framework is experimentally evaluated using two benchmark educational learning analytics datasets, such as OULAD and EdNet. These datasets provide the necessary learner interaction, assessment, and activity data required to evaluate the proposed learner knowledge estimation, personalized optimization, adaptive agent coordination, trust evaluation, and academic-integrity analysis modules. OULAD contains learner demographic information, valuation results, module information, and Virtual Learning Environment (VLE) interactions, which enable learner progress analysis and behavioural pattern analysis. EdNet provides large-scale learner-question interaction data, which includes question attempts and response histories, that supports concept-level knowledge estimation and personalized learning analysis.

The EdNet and OULAD primarily provide learner interaction, assessment, and educational-process information. These datasets are therefore used to construct learner profiles, estimate knowledge states, and evaluate personalized learning performance. For generated educational content, hallucination, trust, explainability, and academic-integrity evaluation, the dataset records are supplemented with corresponding reference answers, learning objectives, and required educational information where applicable. These references provide the basis for checking factual consistency, content relevance, explanation quality, and unsupported information. The evaluation criteria are applied consistently across all responses, while dataset-derived learner information is kept separate from the ground-truth information used for content-level validation.

TABLE V. Description of the OULAD and EdNet Datasets used for Training and Testing the LITE-Trust

Dataset	Category	Total	Training (80%)	Testing (20%)
OULAD	Students	32,593	26,074	6,519
	VLE interactions	10,655,280	8,524,224	2,131,056
EdNet	Students	784,309	627,447	156,861
	Interactions	131,441,538	105,153,230	26,288,307

Table V represents the dataset descriptions, which include total samples and training and testing samples. For both datasets, the data were split into 80% training and 20% test samples at the level of learners so that no interactions from the same learner would appear in both train and test sets. For the purposes of the experiments described in this study, the evaluation was carried out in five independent runs with random seeds 11, 22, 33, 44, 55, while preserving the same learner-level split protocol across all runs. The values reported for the experiments are the mean ± standard deviation over the five runs. The test set was kept hidden during training, parameter tuning, and optimization.

TABLE VI. Hardware and Software Configuration used for Experimental Evaluation

Hardware Configuration	
Processor	Intel Core i7-12700H, 14 cores
Storage	512 GB SSD
RAM	16GB
Software Configuration	
Operating system	Ubuntu 22.04 LTS
Programming Language	Python 3.9
Development environment	Pycharm
Embedding API	Text-embedding-3-small

TABLE VII. Parameter Configuration and Experimental Settings of the LITE-Trust

Parameters	Value
LLM	GPT-40-mini (external API)
Embedding model	Text-embedding-3-small (external API)
Agent activation	Context-adaptive
Tokenization	Standard tokenizer
Retrieval	Dense-vector similarity search
Trainable parameters	1.92 M (LITE-Trust components only)
Prompt strategy	Compact context-aware prompt
KSE weight constraint	$\sum_j w_j = 1$
ELRA population size	30
ELRA exploration coefficient	0.5
ELRA exploitation coefficient	0.5
ELRA maximum iterations	100
Trust threshold	0.70
Agent activation threshold	0.70
Convergence tolerance	0.001

Table VI represents the system configuration, and Table VII illustrates the parameter configuration of the LITE-Trust framework. The reported 1.92 M trainable parameters are for the locally deployed LITE-Trust components only; the externally accessed models (GPT-4o mini and text-embedding-3-small) are not included. The costs associated with the external LLM and embedding services, such as network/API latency and generated cost, are treated separately from the computational costs of the locally deployed LITE-Trust components. Here, the activation threshold was fixed at 0.70 across all experiments to maintain a stable relevance criterion for selective agent execution.

B. Evaluation Metrics

1) Educational performance

a) Knowledge gain: It is generally denoted by the improvement in the learner's knowledge from the pre-learning assessment to the post-learning assessment, which is expressed as,

$$G_K = \mathfrak{H}_{post} - \mathfrak{H}_{pre} \tag{45}$$

where, $\mathfrak{H}_{post}$ represents the pre-learning assessment score, and $\mathfrak{H}_{pre}$ represents the post-learning assessment score.

b) Prediction accuracy: It is given by the ratio of the number of true responses identified by the learners, which is defined as,

$$ACC = \frac{T_P+T_N}{T_P+T_N+F_P+F_N} \tag{46}$$

where, T_P, T_N, F_P, F_N represents true positive, true negative, false positive, and false negative, respectively.

c) F1-Score: It is used to measure the balance between precision and recall, which is expressed as,

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{47}$$

d) Recommendation precision and recall: Precision is used to measure the percentage of relevant recommendations among the selected resources, while recall is used to measure the percentage of relevant resources that were effectively recommended.

$$RP = \frac{T_{PR}}{T_{PR}+F_{PR}}, RR = \frac{T_{PR}}{T_{PR}+F_{NR}} \tag{48}$$

where, $T_{PR}, F_{PR}, and\ F_{NR}$ represent recommended relevant resources, recommended irrelevant resources, and relevant but unrecommended resources, respectively.

e) Student satisfaction: It is used to measure the average learner satisfaction according to the adopted questionnaire or Likert-scale responses, which is defined as $S_S = \frac{1}{m}\sum_{i=1}^{m} \mathcal{s}_i$. Where $\mathcal{s}_i$ represents the learner satisfaction score, and m represents the participating learners' count.

2) Trustworthiness

a) Trust score: It is used to measure the overall trustworthiness of the produced educational outcomes by incorporating factual correctness, confidence, curriculum alignment, explanation quality, and educational completeness. The trust score is expressed as, $T_S = \sum_{l=1}^{m} w_l T_l,\ \sum_{l=1}^{m} w_l = 1$ Where T_l represents each trust-relevant measure and w_l represents its respective weight value.

b) Hallucination Rate: It is used to measure the percentage of produced responses containing unsupported or inaccurate information, which is expressed as,

$$H_R = \frac{n_h}{n_g} \times 100 \tag{49}$$

where, n_h represents the count of responses including hallucinated information and n_g represents the total count of generated responses. A reduced hallucination rate indicates the highest trustworthiness. To calculate the relative enhancement in hallucination control, the hallucination reduction is formulated as,

$$HR_{rd} = \frac{H_{R,base}-H_{R,LITE}}{H_{R,base}} \times 100 \tag{50}$$

where, HR_{rd} denotes the hallucination reduction, $H_{R,base}$ denotes the hallucination rate of the LLM-based tutoring baseline, and $H_{R,LITE}$ denotes the hallucination rate of LITE-Trust under the same evaluation protocol. Hence, the hallucination reduction demonstrates the relative decrease in hallucinated responses obtained by Trust compared with the selected baseline.

c) Factual consistency: Factual consistency refers to the percentage of responses that are factually consistent with educational knowledge or reference information, which is expressed as,

$$F_C = \frac{n_c}{n_v} \times 100 \tag{51}$$

where, n_c represents the count of statements that are factually consistent and n_v represents the total count of the assessed statements. The higher the factual consistency, the better the educational performance is achieved.

For the trust assessment, the EdNet and OULAD datasets provide the context of learner interaction, assessment, and the educational process, but are not used as direct grounds for answers. The produced responses are scored against the ground-truth answers and corresponding educational question-answer information provided with the evaluated learning tasks, which serve as reference information for factuality and hallucination evaluation. The factual correctness is determined by factual statements extracted from the generated responses and compared to the reference information. In addition, the alignment to the curriculum is evaluated based on the target concept and learning objective of the given prompt. The hallucination probability is inferred from the response's unsupported or incorrect statements identified against the reference information. The confidence score is obtained from the model confidence assessment, while the explanation quality is based on the response's relevance, completeness, clarity, and overall consistency. Finally, the educational completeness is defined by the response's ability to fully incorporate the required elements of the given prompt.

3) Explainability

a) Explanation fidelity: It is generally used to measure the accuracy with which the obtained explanation reflects the

factors leading to an educational decision. It is expressed as follows,

$$E_F = \frac{n_c}{n_e} \times 100 \quad (52)$$

where, n_c represents the number of explanations that are consistent with the decision-making process and n_e represents the total number of explanations evaluated.

b) Explanation quality: It is generally used to estimate the explanation quality according to its relevance, completeness, clarity, and consistency. $E_Q = \frac{1}{n}\sum_{i=1}^{n} q_i$, where q_i represents the quality score of the explanation i, and n denotes the count of evaluated explanations.

c) User interpretability: it is used to assess learners' ability to understand and interpret the obtained explanations, which is expressed as $I_U = \frac{1}{n}\sum_{i=1}^{n} u_i$, where u_i represents the interpretability rating generated by the learner i, typically assessed using a Likert-scale evaluation, where higher scores demonstrate greater user interpretability.

For the purpose of explainability evaluation, the fidelity of the produced explanation to the factors that contributed to the corresponding educational decision is determined. Explanation quality is measured in terms of relevance, completeness, clarity, and consistency of the derived explanation. User interpretability is then derived from the learner-oriented interpretability metric as specified in the evaluation procedure. The evaluation is conducted using the same criteria and scoring process for all obtained explanations; hence, no separate expert-panel or multi-rater evaluation is necessary, and inter-rater reliability is not determined. The fidelity, explanation quality, and interpretability measures reported in the study are automated, criterion-based indicators designed to provide a consistent and reproducible assessment of generated explanations. They quantify properties such as consistency with the generated response, relevance, completeness, clarity, and alignment with the associated learning content. These measures are therefore intended as proxy indicators of explanation quality.

4) Efficiency

a) Inference latency: It is used to measure the average execution time taken by the model to generate an output, which is formulated as,

$$L_I = \frac{1}{n}\sum_{i=1}^{n} t_i \quad (53)$$

where, t_i represents the inference time for interaction i.

b) Memory consumption: It is used to measure the memory consumed by the model during execution. Therefore, low memory utilization indicates more effective utilization of system resources and supports placement in resource-constrained educational environments.

c) Number of model parameters: It is used to estimate the total number of trainable parameters of the LITE-Trust model, which reflects the memory consumption and model complexity, where a lower parameter count indicates that the model is lightweight.

d) FLOPs: It is used to evaluate the number of floating-point operations required during model inference, which reflects the computational complexity of the LITE-Trust architecture, while fewer FLOPs indicate lower computational complexity.

e) Computational cost: It is used to measure the computational burden of the system, with lower values indicating more computational efficiency and lighter algorithms.

f) Response time: It is used to compute the overall processing time taken from receiving the request from the user until the moment of providing the final response, which is defined as,

$$R_T = t_o - t_i \quad (54)$$

where, t_o, and t_i represent the request delivery, and request response times, respectively.

To improve the objectivity and reproducibility of the trust model, all trust components are determined using predefined evaluation criteria and are kept fixed across the experiments. Factual correctness and hallucination are evaluated against the corresponding ground-truth answers and educational reference information, while curriculum alignment is assessed with respect to the target concept and learning objective of each task. Confidence is obtained from the model's confidence assessment. Explanation quality is evaluated according to relevance, completeness, clarity, and consistency, whereas educational completeness reflects whether the required instructional elements of the task are adequately covered. These criteria are applied uniformly to all evaluated responses rather than being adjusted according to the obtained performance. The weights assigned to the individual trust components are fixed before the experimental evaluation and are applied identically to all datasets, learner profiles, and generated responses. The weight selection is based on the relative importance of factual reliability, instructional relevance, and response quality for educational recommendation. No weight is modified based on the performance of an individual model or test sample. This fixed-weight protocol prevents post-hoc adjustment of the trust score and improves reproducibility across experiments.

The reliability of the trust assessment is further examined by evaluating the individual trust components separately from the overall trust score. Factual consistency and hallucination rate are evaluated against reference educational information, while curriculum alignment and educational completeness are assessed against the predefined learning objectives and task requirements. Explanation quality is evaluated independently using its relevance, completeness, clarity, and consistency criteria. The component-level results are then compared with the aggregated trust score to verify that high trust values correspond to responses with stronger factual reliability, lower unsupported content, appropriate curriculum coverage, and better educational completeness.

To improve the reliability of the knowledge-gain evaluation, the pre-learning and post-learning assessments are compared under consistent learner and assessment conditions.

The same learner profiles, target concepts, assessment criteria, and evaluation protocol are maintained across the compared methods. Learner characteristics and prior knowledge are considered when interpreting the observed improvement, while temporal progression, repeated exposure, and differences in learner participation are controlled as far as permitted by the available data.

C. Overall Performance Comparison

The LITE-Trust is compared with existing approaches such as conventional ITS [15], conventional multi-agent systems [10], LLM-based tutoring [6], non-optimized multi-agent systems [12], optimization-based approach [20], and adaptive learning models [7], to evaluate its performance. These baselines were selected as they represent major paradigms of the educational system addressed by LITE-Trust, namely, rule-based tutoring, multi-agent coordination, LLM-assisted learning, adaptive learning modelling, and optimization-driven personalization. Each baseline is evaluated on the same EdNet and OULAD datasets, with the same train/test partitions, hardware configuration, and evaluation metrics as LITE-Trust, using comparable learner-level partitions of the data to ensure validity of comparison, with variations specific to each method and their implementation details retained.

For a fair comparison, all baseline methods and the proposed LITE-Trust framework were evaluated under the same experimental setting, including the same datasets, learner profiles, input information, evaluation protocol, and computing environment. The same experimental scenarios and stopping criteria were maintained for all methods. Each baseline was implemented using its defined methodological characteristics, while the proposed framework used the same available input and evaluation conditions. Model-specific parameters, prompts, and configuration settings were kept fixed within each method and were not selectively adjusted to favor the proposed approach. This ensures that the reported differences are primarily attributable to the methodological differences between the compared approaches rather than changes in the experimental conditions.

The results in Tables VIII–XV are presented as descriptive performance comparisons, with mean values and standard deviations used to indicate central performance and variability across experimental runs. Formal statistical testing was applied to the optimizer-level comparison because ELRA was specifically evaluated against alternative optimization strategies. Nevertheless, to strengthen the statistical reliability of the broader baseline comparisons, confidence intervals and/or appropriate pairwise statistical tests can additionally be reported for the headline metrics. Differences such as 95.6% versus 90.1% are interpreted as observed performance improvements rather than statistically significant differences unless supported by the corresponding statistical test.

TABLE VIII. ANALYSIS OF EDUCATIONAL PERFORMANCE ON EDNET AND OULAD DATASET

Model	EdNet			OULAD		
	KG (%)	Accuracy (%)	F1 (%)	KG (%)	Accuracy (%)	F1 (%)
Conventional ITS	12.7±0.5	85.5±0.5	85.8±0.5	11.9±0.6	85.1±0.5	85.1±0.5
Conventional multi-agent system	13.6±0.5	87.5±0.6	86.0±0.7	12.9±0.6	87.0±0.5	86.1±0.7
LLM-based tutoring	14.5±0.6	87.6±0.5	87.9±0.6	13.4±0.7	86.9±0.4	87.1±0.6
Non-optimized multi-agent system	15.2±0.4	89.1±0.5	88.7±0.5	14.2±0.6	88.8±0.6	88.3±0.6
Adaptive Learning model	15.3±0.7	89.2±05	89.2±0.5	14.5±0.5	88.9±0.6	88.3±0.5
Optimization-based approach	16.5±0.4	90.1±0.5	89.9±0.6	15.1±0.5	89.8±0.5	89.0±0.5
LITE-Trust	**18.8±0.5**	**95.6±0.2**	**94.5±0.4**	**17.9±0.4**	**94.8±0.3**	**93.5±0.4**

Table VIII explains the numerical results of educational performance on the EdNet and OULAD datasets. The LITE-Trust achieves the highest performance with an accuracy of 95.6%, an F1-score of 94.5%, knowledge gain of 18.8% on the EdNet dataset, and 94.8%, 93.5%, and 17.9% on OULAD, respectively. The optimization-based approaches achieve only 90.1% accuracy and 89.9% F1-score on EdNet, and 89.8% accuracy and 89.0% F1-Score on the OULAD dataset. This enhancement is obtained by integrating KSE-based concept-level knowledge estimation with ELRA-based educational optimization, which enables decisions to be modified to the learner's current knowledge and needs. Unlike existing approaches, which generally focus on a single capacity, such as tutoring, prediction, adaptation, multi-agent processing, or optimization, LITE-Trust collectively models learner state and optimizes personalized educational configurations. The reported results are expressed as mean ± standard deviation to indicate the performance variability across individual runs. For knowledge gain evaluation, the learner interactions in each learning episode are divided into temporally ordered pre-learning and post-learning windows relative to the personalized learning process. Specifically, the pre-learning window comprised the assessment interactions preceding the recommended learning process, whereas the post-learning window comprised the assessment interactions following it. Knowledge gain (KG) is calculated from the enhancement in mean assessment performance between these two windows. These results are gained from KSE-driven recommendations and ELRA-guided optimized agent coordination, which minimize redundant educational actions and enhance learner-specific resource selection.

Table IX explains the performance of trustworthiness on EdNet and OULAD datasets. The LITE-Trust system achieves 94.5%/93.8% trust, 3.1%/3.4% hallucination, and 96.4%/95.6% factual consistency, demonstrating superior performance compared with the existing approaches. This achievement was obtained by integrating student profile

construction, KSE, ELRA, personalized agent coordination, and trust verification, which preserves learner-specific, context-guided, and factually interpretable decisions. Existing approaches primarily focus on individual capabilities without integrating learner profiles with continuous trust evaluation, which results in lower trust and higher hallucination.

TABLE IX. ANALYSIS OF TRUSTWORTHINESS ON EDNET AND OULAD DATASET

Model	EdNet			OULAD		
	Trust (%)	Hallucination (%)	FC (%)	Trust (%)	Hallucination (%)	FC (%)
Conventional ITS	82.1±0.7	7.8±0.5	86.3±0.7	81.1±0.8	8.0±0.6	85.0±0.8
Conventional multi-agent system	82.9±0.8	7.5±0.5	86.7±0.7	81.8±0.8	7.8±0.5	86.1±0.7
LLM-based tutoring	84.5±0.8	7.0±0.5	88.2±0.6	83.6±0.9	7.3±0.5	86.9±0.7
Non-optimized multi-agent system	85.5±0.7	6.8±0.5	87.5±0.6	84.5±0.8	7.1±0.5	87.2±0.7
Adaptive Learning model	86.4±0.7	6.1±0.4	89.5±0.6	85.0±0.7	6.5±0.6	88.3±0.6
Optimization-based approach	87.1±0.6	5.9±0.4	89.8±0.5	86.1±0.7	6.3±0.6	89.1±0.6
LITE-Trust	**94.5±0.4**	**3.1±0.3**	**96.4±0.4**	**93.8±0.5**	**3.4±0.3**	**95.6±0.5**

TABLE X. ANALYSIS OF EXPLAINABILITY ON EDNET AND OULAD DATASET

Model	EdNet			OULAD		
	Fidelity (%)	Quality (%)	Interpretability (%)	Fidelity (%)	Quality (%)	Interpretability (%)
Conventional multi-agent system	80.5±0.8	79.3±0.9	76.9±0.7	79.8±0.7	78.5±0.7	76.2±0.8
Conventional ITS	82.3±0.7	80.6±0.8	78.7±1.0	81.6±0.9	79.8±0.8	77.8±0.8
Non-optimized multi-agent system	84.3±0.8	83.0±0.8	81.2±0.9	83.7±0.9	82.0±0.9	80.4±0.7
LLM-based tutoring	85.3±0.6	83.1±0.8	81.9±0.8	84.2±0.8	82.4±0.7	80.9±0.7
Optimization-based approach	86.5±0.7	84.5±0.8	83.4±0.8	85.1±0.8	84.2±0.9	82.3±0.8
Adaptive Learning model	86.9±0.7	85.6±0.7	84.1±0.8	85.8±0.8	85.0±0.8	83.5±0.6
LITE-Trust	**95.0**±0.4	**94.1**±0.5	**92.6**±0.5	**94.6**±0.5	**93.4**±0.5	**92.1**±0.5

Table X explains the explainability performance on EdNet and OULAD datasets. The LITE-Trust model achieves the highest fidelity of 95.0%/94.6%, explanation quality of 94.1%/93.4%, and interpretability of 92.6%/92.1% across two datasets. The highest results were achieved by implementing context-aware generation and validation of explanations, where the system combines learner-specific evidence and decision outputs before generating the final explanations. Subsequently, the produced explanations remain close to the decisions actually made and are detailed enough to be informative for the learners. In contrast, existing methods often generate generalized explanations with restricted alignment to the learner-specific personalized decisions, which leads to lower fidelity, quality of explanation, and interpretability.

TABLE XI. ANALYSIS OF COMPUTATIONAL EFFICIENCY ON EDNET AND OULAD DATASET

Model	EdNet			OULAD		
	Latency (ms)	Memory (MB)	Parameters (M)	Latency (ms)	Memory (MB)	Parameters (M)
Adaptive Learning model	16.3	92.5	2.19	17.4	93.4	2.19
Conventional ITS	17.4	95.4	2.71	18.2	98.2	2.71
Non-optimized multi-agent system	18.4	98.7	3.08	19.5	99.7	3.08
Optimization-based approach	19.4	109.5	3.75	19.8	110.1	3.75
Conventional multi-agent system	19.5	105.8	3.43	20.4	106.9	3.43
LLM-based tutoring	20.2	118.1	6.20	21.5	119.8	6.20
LITE-Trust	**10.5**	**80.5**	**1.92**	**9.5**	**81.3**	**1.92**

Table XI depicts the computational efficiency of proposed and existing approaches on the EdNet and OULAD datasets in terms of latency, memory usage, and parameter count. LITE-Trust achieves the lowest latency of 10.5 ms and 9.5 ms, memory usage of 80.5 MB and 81.3 MB, and parameter count of 1.92 M on EdNet and OULAD, respectively. Compared with existing methodologies, the proposed LITE-Trust system minimizes computational needs while preserving a lightweight architecture. The lowest latency demonstrates faster execution, while the reduced number of memory and parameters proves

the method's lightness. In addition, the similar results for different datasets signify that the efficiency of the LITE-Trust model is not tied to a particular dataset.

D. Earning Personalization Performance

Table XII illustrates the learning personalization performance of presented and existing works in terms of learner adaptation, knowledge-gap identification, recommendation accuracy, and learning-path quality. LITE-Trust system achieves the highest scores of 94.1%, 93.2%, 94.0%, and 92.9%, respectively. This enhancement is achieved by the combined effect of the presented approach, where the student profile construction extracts individual learner characteristics, KSE preserves the evolving knowledge state, ELRA improves learning configurations, and responsive multi-agent architecture allocates learner-specific instructional tasks. The explainability engine then evaluates and transfers the rationale behind the produced learning decisions, while the trust-based strategy makes recommendations for trustworthy educational information. Unlike existing approaches, which generally deploy less integrated personalization strategies and generate restricted coordination among learner profiling, knowledge evaluation, optimization, recommendation, and decision validation. Therefore, they are less efficient in adjusting to changing learner needs, identifying specific knowledge gaps, and constructing high-quality personalized learning paths.

TABLE XII. ANALYSIS OF LEARNING PERSONALIZATION PERFORMANCE

Model	Learner Adaptation (%)	Knowledge Gap Identification (%)	Recommendation Accuracy (%)	Learning Path Quality (%)
LLM-based tutoring	81.2±0.7	79.9±0.8	80.5±0.7	78.7±0.8
Conventional multi-agent system	83.6±0.6	81.6±0.7	82.7±0.6	80.6±0.7
Conventional ITS	84.5±0.7	82.6±0.7	83.9±0.7	81.7±0.9
Non-optimized multi-agent system	85.3±0.6	83.4±0.6	84.2±0.6	82.6±0.7
Adaptive Learning model	86.4±0.5	84.2±0.7	85.1±0.7	83.4±0.6
Optimization-based approach	87.2±0.5	85.7±0.6	86.7±0.6	84.7±0.6
LITE-Trust	**94.1**±0.4	**93.2**±0.5	**94.0**±0.4	**92.9**±0.5

E. Trust and Hallucination Analysis

Table XIII depicts the trust and hallucination analysis of the proposed and existing approaches. The LITE-Trust model achieves the highest trust of 94.15% and hallucination reduction of 54.6%, which demonstrates the enhanced reliability of the produced educational outcomes. The overall trust score is calculated as the average of the dataset-specific trust scores obtained on EdNet and OULAD. The hallucination reduction values are calculated using Eq. (33), with the LLM-based tutoring approach as the baseline. The results show a gradual improvement in hallucination reduction from the LLM-based tutoring method to the conventional multi-agent, conventional ITS, non-optimized multi-agent, adaptive learning, and optimization-based approaches, while LITE-Trust achieves the highest reduction.

TABLE XIII. TRUST AND HALLUCINATION ANALYSIS

Model	Trust Score (%)	Hallucination Reduction (%)
LLM-based tutoring	80.8±0.9	18.6±0.8
Conventional multi-agent system	82.5±0.8	22.5±0.8
Conventional ITS	83.7±0.7	25.3±0.7
Non-optimized multi-agent system	84.9±0.6	29.3±0.8
Adaptive Learning model	85.8±0.6	34.8±0.7
Optimization-based approach	86.8±0.5	39.8±0.6
LITE-Trust	**94.15±0.4**	**54.6±0.5**

The enhancement is qualified by the incorporation of student profile construction, KSE, ELRA, personalized multi-agent coordination, explainability engine, and trust-based validation approaches. These subsystems jointly support learner-specific decision generation, evidence-based recommendations, explanation evaluation, and controlled response generation. In contrast, the existing methods generally provide relatively restricted strategies for collectively evaluating generated content, learner-specific decisions, and explanatory consistency. The higher hallucination reduction achieved by LITE-Trust therefore indicates improved control of potentially unsupported responses, contributing to higher overall trust in the generated educational outcomes.

F. Explainability Analysis

Table XIV presents the explainability performance of the proposed and existing approaches in terms of quality and transparency. Table XV represents the examples of

explanations and recommendations. The LITE-Trust model achieves superior performance of 94.1% explanation quality and 93.8% transparency, which shows that it provides informative and transparent explanations to the learner. The quality of the explanations and the transparency of the recommendations in the explanations were assessed using an automated scoring protocol. The explanations were scored and compared based on relevance, adequacy, clarity, consistency, and correspondence with the recommendation. For each explanation, scores were calculated based on these criteria and converted into percentages for comparison between the recommendation models. Additionally, the qualitative examples demonstrate that LITE-Trust is able to link the learner context to the recommended learning resources along with presenting an explanation for the recommendation outputs. This is made possible due to the combination of comprehensive learner profiling, knowledge-state representation, ELRA-based decision optimization, adaptive multi-agent coordination, and explainability engine mechanisms, which together facilitate exposing the recommendation factors. On the other hand, the compared solutions provide only the recommendation output with little information about the factors influencing the outcome, thus being less transparent and informative in terms of explanation.

To ensure objective evaluation, fidelity, explanation quality, and interpretability are calculated using predefined evaluation criteria applied consistently across all generated responses. Fidelity measures the extent to which the explanation accurately reflects the evidence and reasoning associated with the generated response. Explanation quality evaluates relevance, completeness, clarity, and consistency of the explanation, while interpretability assesses whether the explanation can be readily understood and traced to the corresponding learning content and response. The evaluation criteria and scoring procedure are kept fixed across all experiments to avoid subjective variation. The reported percentages represent normalized scores aggregated across the evaluated responses. No post-hoc adjustment of scores is performed based on the obtained results. Where human evaluation is included, responses are independently assessed using the same predefined criteria by the designated evaluators, and the final score is obtained from their assessments.

TABLE XIV. EXPLAINABILITY ANALYSIS

Model	Explanation Quality (%)	Transparency (%)
LLM-based tutoring	80.5±0.7	79.6±0.7
Conventional multi-agent system	82.0±0.6	81.4±0.7
Conventional ITS	84.5±0.7	82.8±0.8
Non-optimized multi-agent system	85.6±0.6	84.7±0.7
Adaptive Learning model	86.9±0.5	85.4±0.6
Optimization-based approach	87.5±0.5	86.1±0.6
LITE-Trust	**94.1±0.4**	**93.8±0.5**

TABLE XV. EXAMPLE OF RECOMMENDATION AND EXPLANATIONS

Student Context	Recommendation	Generated Explanation
Low performance in algebra; good performance in arithmetic, fair performance in equations	Intermediate algebra learning resource	The learner is adequately proficient in arithmetic, but lacks knowledge in solving equations. Hence, it is recommended to study intermediate algebra to improve equation-solving skills
Poor understanding of probability, moderate performance in statistics fundamentals	Probability learning resource-intermediate level	The learner lacks an adequate understanding of the idea of probability. The recommended module was chosen based on the learner's current knowledge and the identified learning needs.

G. Lightweight Performance Analysis

Table XVI depicts the existing non-optimized multi-agent method with the full-agent, selective multi-agent, and overall LITE-Trust configurations. The evaluation in Table XVI is conducted on the OULAD dataset for the learner outcome prediction task. Full-agent LITE-Trust has more overhead since all agents are activated, while selective activation only triggers relevant agents according to educational requirements. The parameter counts presented in Table XVI relate only to the trainable parameters in the proposed LITE-Trust components, while the external GPT-4o mini model remains constant across all configurations of LITE-Trust. The difference between 2.46M and 3.35M in the full-agent configuration stems from the activation of all task-specific agents and their corresponding processing layers. In the selective multi-agent configuration, only the agents relevant to the specified requirements are engaged, thus accounting for a lower count of trainable parameters at 2.46 M. The overall LITE-Trust configuration further decreases the parameter count to 1.92 M due to parameter sharing and the elimination of task-specific components within the integrated pipeline. Therefore, the variation in parameter counts originates from the proposed configurations rather than the external large language model. The 94.8% accuracy reported for the overall LITE-Trust configuration corresponds to the OULAD-specific result in Table VIII, whereas the 95.6% accuracy in Table VIII represents the separate EdNet evaluation.

TABLE XVI. LIGHTWEIGHT PERFORMANCE ANALYSIS

Method	Parameters (M) ↓	Latency (ms) ↓	Memory (MB) ↓	Accuracy (%) ↑
Non-optimized multi-agent system	3.08	18.4	98.7	88.7
Full multi-agent LITE-Trust	3.35	16.9	102.1	95.4
Selective multi-agent LITE-Trust	2.46	11.7	89.5	95.2
Overall LITE-Trust	**1.92**	**9.5**	**81.3**	**94.8**

The reported 1.92 M parameters represent only the locally implemented components of the proposed framework and exclude the externally accessed GPT-4o mini and embedding model. Therefore, the parameter count should not be interpreted as the complete deployed-system size. To provide a more realistic assessment of lightweight deployment, the evaluation considers end-to-end factors including local computational resources, API response latency, network communication overhead, external inference cost, and memory usage. The framework is therefore characterized as lightweight primarily with respect to its locally deployed components and selective agent execution, rather than as a parameter-light representation of the entire external AI service stack.

The overall LITE-Trust reduces computational costs by several means, such as compact learner and knowledge-state representations, limited ELRA search, selective agent activation, and efficient algorithms for processing explanations and trust. Selective agent activation contributes to efficiency by reducing unnecessary agent execution; however, the overall end-to-end enhancement cannot be attributed solely to reducing the number of active agents. Instead, the results indicate the agent-level reduction works together with the optimization of the complete pipeline, such as KSE, ELRA, retrieval, trust, and XAI processing. Therefore, selective activation provides an important agent-level efficiency benefit, while the overall lightweight performance results from the combined optimization of the entire LITE-Trust pipeline.

H. ELRA Optimization Performance

To ensure consistency in comparison, GA, PSO, ACO, LRA, and ELRA have been evaluated using the same population size, equal to 30, the same learner-specific fitness function, and the same experimental scenario with five independent runs using random seeds 11, 22, 33, 44, and 55. The value of 30 is selected for population size on the basis of the performed sensitivity analysis to balance the solution quality and computational effort. All the configurations used in the experiments can be considered reasonable as they conform to standard educational performance measures and use the specific search and update mechanisms of the respective algorithms.

Table XVII depicts the comparison of the proposed and conventional optimization algorithms, namely GA, PSO, ACO, and LRA, in terms of convergence, fitness, standard deviation, computational time, agent activation, and learning performance. ELRA achieves faster convergence, higher fitness, the lowest standard deviation, lower computational time, fewer activated agents, and the best learning performance among the compared optimizers. The lowest standard deviation means high consistency of the results of independent runs. These advantages were achieved due to ELRA's learner-state-guided initialisation, constrained candidate generation and relevance-based optimization, and quality-and diversity-based selection, which aims to facilitate optimization by focusing the search on promising learner-specific configurations and avoiding redundant exploration. The overall comparison among five optimizers was statistically significant based on the Friedman test ($p \approx 0.001$), which indicates significant differences in optimizer performance across the evaluated methods.

The direct comparison with conventional LRA further demonstrates the contribution of the ELRA modifications. Under the same population size, learner-specific fitness function, experimental scenario, and independent random seeds, ELRA improves the best fitness from 0.89 obtained by LRA to 0.93, corresponding to a relative improvement of approximately 4.5%. ELRA also reduces convergence iterations from 38 to 28 (26.3% reduction), standard deviation from 0.025 to 0.012 (52.0% reduction), computational time from 15.4 ms to 10.8 ms (29.9% reduction), and average activated agents from 3.9 to 2.7 (30.8% reduction). Learning performance increases from 91.2% to 94.8%, an absolute improvement of 3.6 percentage points. These results indicate that the learner-state conditioning, trust-aware candidate filtering, and relevance-based search mechanisms provide measurable benefits beyond the conventional LRA search strategy.

TABLE XVII. ELRA OPTIMIZATION PERFORMANCE

Optimizer	Convergence iterations ↓	Best Fitness ↑	Standard Deviations ↓	Computational Time (ms) ↓	Activated Agents ↓	Learning Performance (%) ↑
GA	50	0.86	0.035	21.5	4.8	89.5
PSO	47	0.87	0.031	18.9	4.6	89.6
ACO	44	0.88	0.029	17.8	4.2	90.5
LRA	38	0.89	0.025	15.4	3.9	91.2
ELRA	**28**	**0.93**	**0.012**	**10.8**	**2.7**	**94.8**

I. Ablation Study

Table XVIII depicts the ablation study results, which highlight the contributions made by each component of the proposed LITE-Trust. Excluding ELRA reduces the learning pathway quality from 94.8% to 89.5% since the learner's KSE is not transformed into an optimized learning configuration, resulting in lower resource and path utility. This confirms ELRA's role in converting learner-state information into personalized educational configurations. Excluding the trust engine lowers trust by 9.8% because the produced recommendations are no longer validated against the reliable educational information repository via the trust-aware validation mechanism. This degradation reflects the role of trust evaluation in improving the reliability of educational decisions. Removing the explainability engine decreases the explanation quality from 94.6% to 78.3%, representing a 16.3% reduction, while slightly reducing latency from 9.5 ms to 8.8 ms. This trade-off is expected because removing XAI eliminates explanation-processing overhead but reduces transparency and interpretability. Excluding dynamic agent selection increases the latency from 9.5 ms to 16.5 ms, representing a 73.7% increase, because unwanted agents are activated instead of directly employing only the relevant agents. These results demonstrate that LITE-Trust balances multiple objectives rather than maximizing every metric simultaneously, with XAI improving interpretability and agent selection enhancing computational efficiency. Finally, the overall performance of the complete LITE-Trust achieves 94.8% accuracy, 95.2% trust score, 94.6% XAI quality, 9.5 ms latency, and 94.8% learning path quality, since the student profile construction accurately represents the learners' features. KSE maintains the required knowledge state, ELRA determines the best configuration, dynamic coordination activates only necessary agents, and the trust and XAI engine enhance the transparency of decisions and recommendations. Therefore, the observed degradation after removing each component is consistent with its intended function, which confirms the complementary contribution of ELRA, trust evaluation, XAI, and dynamic agent selection to personalized, reliable, transparent, and efficient learning.

TABLE XVIII. ABLATION STUDY OF LITE-TRUST SYSTEM

Model	Accuracy (%)	Trust Score (%)	XAI Quality (%)	Latency (ms)	Learning-path Quality (%)
Model A: without ELRA	91.7	92.5	91.6	14.3	89.5
Model B: Without Trust Engine	93.8	85.4	91.9	9.7	93.0
Model C: Without XAI	94.0	94.2	78.3	8.8	93.6
Model D: Without Agent Selection	94.2	94.0	93.6	16.5	94.1
Model E: Overall LITE-Trust	**94.8**	**95.2**	**94.6**	**9.5**	**94.8**

To further clarify the contribution of individual modules, the ablation analysis is organized as controlled component-wise comparisons. Each variant removes one module while keeping the dataset, learner profiles, experimental conditions, and remaining framework components unchanged; changes in accuracy, trust, explainability, latency, and learning-path quality are compared with the complete model. The resulting LITE-Trust framework. This analysis identifies the principal contribution of each component while minimizing the influence of changes in other experimental factors. The observed improvements are interpreted as component-level contributions; interaction effects among multiple modules are considered a limitation of the current ablation design and are not claimed as independently established causal effects.

J. Sensitivity and Convergence Analysis

Table XIX shows that increasing the population size from 10 to 30 raises the fitness from 0.90 to 0.93 and the learning performance from 92.2% to 94.8%, while further increasing it to 40 provides only a marginal improvement to 94.9%. Similarly, increasing the number of iterations beyond 0 provides only marginal improvement in learning performance. Therefore, the selected configuration represents a performance-efficiency trade-off, where a population size of 30 and 60 iterations provides nearly the same performance as larger configurations while avoiding additional computational cost. Thus, the balanced configuration was optimized for both educational effectiveness and computational performance, which reached the maximum fitness of 0.93 and a learning performance of 94.8%. These results suggest that ELRA is fairly robust to parameter variations, as moderate changes to the population size and iteration number affected the outcomes within a reasonable range. More importantly, population size, iteration number, and by extension, the optimization process itself are predefined computational constraints, not learner-specific variables. Hence, while ELRA extends the lightweight optimization principles of LRA, its educational adaptation introduces a restricted set of control parameters to control optimization and agent activation, while retaining the lightweight optimization philosophy of the original LRA. The weighting strategy is assessed using sensitivity analysis. Table XX below compares the trust, efficiency, learning, and balanced configuration based on attained fitness and learning performance. It can be seen that the balanced configuration had the highest fitness score of 0.93 and the best learning performance of 94.8% and was thus selected as the final ELRA configuration.

To examine whether the selected threshold values are robust, a sensitivity analysis is performed by varying the trust threshold and agent activation threshold across multiple levels while keeping the remaining experimental settings unchanged. The evaluation considers hallucination rate, response quality, agent utilization, latency, and learner performance. The same datasets, learner profiles, prompts, agent pool, and evaluation procedure are maintained for each threshold setting so that the

observed differences can be attributed primarily to threshold variation.

TABLE XIX. SENSITIVITY AND CONVERGENCE ANALYSIS

Parameter	Setting	Convergence iterations ↓	Fitness ↑	Learning performance (%) ↑
Population size	10	33	0.90	92.2
	20	29	0.92	93.7
	30	**27**	**0.93**	**94.8**
	40	27	0.93	94.9
Iterations	20	34	0.89	92.5
	40	29	0.91	94.1
	60	**27**	**0.93**	**94.8**
	80	26	0.92	94.9

TABLE XX. SENSITIVITY ANALYSIS OF ELRA WEIGHTING CONFIGURATIONS

Weighting Configuration	Trust	Efficiency	Learning	Fitness ↑	Learning Performance (%) ↑
Trust-based	0.49	0.20	0.30	0.88	92.5
Efficiency-based	0.20	0.49	0.30	0.89	92.8
Learning-based	0.20	0.29	0.49	0.90	93.4
Balanced	**0.33**	**0.33**	**0.34**	**0.93**	**94.8**

TABLE XXI. STATISTICAL JUSTIFICATION OF THE ELRA AGAINST THE GA, PSO, ACO, AND LRA

Trust threshold	Hallucination rate (%)	Response quality (%)	Agent utilization (%)	Latency (ms) ↓	Learner performance (%)
0.50	7.20	91.40	92.50	182	92.10
0.60	6.10	92.80	89.70	190	93.00
0.70	4.90	94.20	86.30	199	94.10
0.80	3.80	95.60	82.10	211	95.00
0.90	2.90	96.70	76.80	228	95.40

The threshold sensitivity results demonstrate the effect of threshold selection on the reliability and efficiency of the proposed framework. Lower thresholds allow more candidate responses and agents to participate, which can increase agent utilization and potentially reduce computational filtering, but may also permit less reliable responses to proceed. In contrast, higher thresholds impose stricter trust and activation requirements, which can reduce unnecessary agent execution and unsupported responses but may also reject useful responses or increase processing overhead associated with additional validation. The intermediate threshold setting provides a balanced operating point when considering hallucination control, response quality, agent utilization, latency, and learner performance.

K. Statistical Significance Analysis

Table XXI depicts the statistical justification of the ELRA against the GA, PSO, ACO, and LRA. The statistical unit is an independent experimental run, with five runs conducted using seeds 11,22,33,44,55. Learner interactions and independent optimization evaluation are not treated as independent samples. The Friedman test reveals that there are significant differences between the five optimizers used ($x^2 \approx 18.65$, $p \approx 0.001$). The exact two-sided Wilcoxon signed-rank test is applied to the five paired run-level results, yielding $p \approx 0.063$ for the pairwise comparisons. The Nemenyi post-hoc analysis is based on the same five-run rank matrix. The effect sizes vary from 0.64 to 0.83, suggesting that the improvements are substantial. In addition, the 95% confidence interval 0.91-0.93 reflects the stability of ELRA fitness across the five runs. Thus, the statistical analysis uses run-level statistical units and avoids treating correlated learner or interaction observations as independent samples.

TABLE XXII. ANALYSIS OF STATISTICAL SIGNIFICANCE

Metrics	ELRA Evaluation	Result
Statistical unit	Independent experimental runs	5 runs (seeds 11,22,33,44,55)
Friedman test	Overall difference among GA, PSO, ACO, LRA, and ELRA	$x^2(4) \approx 18.65$, $p \approx 0.001$
Wilcoxon signed-rank	ELRA vs. GA/PSO/ACO/LRA	$p \approx 0.063$
Nemenyi post-hoc	ELRA vs. GA/PSO/ACO/LRA	$p > 0.05$
Effect size	ELRA vs. GA/PSO/ACO/LRA	0.83/0.79/0.71/0.64
95% Confidence Interval	ELRA Fitness	0.91-0.93

The statistical results have been clarified to avoid overstating significance (Table XXII). Although ELRA shows better mean performance than the compared optimizers, the pairwise Wilcoxon result of approximately 0.063 and the Nemenyi results above 0.05 do not demonstrate statistically significant superiority at the conventional 0.05 level. Therefore, ELRA is described as showing better observed performance and practical improvement, rather than statistically significant superiority in every pairwise comparison. The overall statistical analysis is interpreted together with effect sizes, confidence intervals, variability across independent runs, and mean performance to provide a more complete assessment of the observed differences.

L. Discussion

The results indicate that the improvements of LITE-Trust come from the complementary interaction of its proposed components, as further enabled by the ablation study; removing any individual module consistently degraded the corresponding performance metrics. ELRA enhances agent selection by leveraging the learner profile and KSE to determine the optimal learning configuration and identify relevant agents, thus restricting the optimization search to the learner-relevant educational configurations and avoiding unwanted exploration of unsuitable learning paths. Moreover, KSE identifies concept-level knowledge gaps and prerequisite mastery, which enables ELRA to allocate learning resources and educational agents based on the learner's actual learning requirements instead of depending on aggregate performance indicators. Trust Evaluation boosts performance by ensuring the factual

consistency of the results through confirming AI-generated educational content before delivery, thus minimizing the inclusion of unsubstantiated information and enhancing the reliability of educational recommendations. Dynamic orchestration facilitates reduced computational costs by limiting the engagement to necessary agents for each learning task, thereby avoiding unwanted agent invocations, which reduces inter-agent communication complexity and lowers processing time and memory consumption. This offers a favourable balance between learning performance and computational demands, as LITE-Trust performs better than all the baseline methods at a lower cost than multi-agent and optimization-based methods. The ablation results further indicate that the combined operation of KSE, ELRA, trust evaluation, explainability, and selective multi-agent orchestration generates more benefits than employing these components individually, which highlights their complementary contributions to personalization, optimization, trustworthiness, explainability, and computational efficiency. In comparison to the existing educational AI methods, which mostly focus on personalization, multi-agent orchestration, optimization techniques, explainability, or trust, LITE-Trust provides an integrated learner-centric approach that combines personalization, orchestration of multiple agents, optimization, explainability, and trust evaluation. In practice, such reduced complexity and selectivity would be critical to support the implementation in contexts where computational resources are limited, but learning personalization, performance, explanation, and trust are needed.

M. Limitations and Future Directions

Although LITE-Trust has shown promising results, there are still several limitations in terms of the current research. Firstly, it should be noted that it is based on a few educational benchmarks, making its ability to generalize questionable. Moreover, there was not enough practical application in a classroom setting, and using these lightweight GenAI components may lead to massive computational overhead. Then, it is hard to quantify the amount of learning gained by students since short-term gains do not always reflect long-term ones. Lastly, automatic trust and hallucination scores may miss many of the nuanced errors in natural language instruction.

The future directions of the research will address these concerns and explore different aspects of educational AI. In particular, they will focus on multimodal learning through speech, imagery, and behavioural data, and instruction via speech and vision. Federated educational AI and edge models will also be used to achieve better privacy and lighter computational needs while building more autonomous agents for education. Furthermore, future research will focus on longitudinal student modeling to understand learning gains better.

V. CONCLUSION

This study addressed the limitation of existing educational AI systems to provide personalized, optimized, trustworthy, explainable, and computationally efficient learning support. The LITE-Trust system integrated student profile construction, KSE, ELRA, dynamic multi-agent orchestration, explainability engine, and trust evaluation into a learner-centric pipeline. First, ELRA enhanced learner-specific educational configurations by leveraging the learner profile and KSE to focus the optimization search on relevant learning paths, resources, and agent configurations while avoiding superfluous optimization steps and agent exploration. Second, the trust-aware and explainable mechanisms improved the measured trust and explanation quality of produced educational outcomes by validating educational information and providing learner-specific rationales. Third, the lightweight multi-agent orchestration mechanism exploited relevance-based selective agent invocations to prevent superfluous agent executions, thereby reducing the measured computational overhead and response latency.

In addition, the behaviour-based academic integrity analyzer complemented the framework by estimating indicators related to academic integrity and AI usage based on learners' interactions, thus providing educators with valuable information without actually measuring these variables directly. Overall, the experimental results demonstrated that the LITE-Trust system achieved higher measured personalization, trust, explainability, and computational efficiency over the compared baseline methods. The ELRA analysis revealed faster convergence, higher fitness, stable optimization, and fewer agent invocations, while the ablation and statistical tests highlighted the complementarity of the integrated components. Thus, the designed approach demonstrated the effectiveness of a combined learner-state modeling, educational optimization, selective multi-agent orchestration, explainability, and trust validation pipeline for the estimated educational learning scenarios. However, the evaluation is limited to the EdNet and OULAD benchmark datasets, and trust assessment is performed through an automated framework-based mechanism rather than long-term human validation. Furthermore, due to the absence of ground-truth labels in the data concerning academic integrity and reliance on AI, the results of the academic integrity analyzer should be regarded as behavioural estimates rather than measurements. Large-scale real-world classroom deployment and comprehensive learner satisfaction measurement were not conducted. These aspects remain important directions for future evaluation.

FUNDING SOURCES

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, Grant No.: KFU265137.

ACKNOWLEDGMENT

The authors are thankful to the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, and Al-Fayha Private College, Jubail, Saudi Arabia, for providing financial assistance.

DATA AVAILABILITY

The datasets used and/or analyzed during the current study are available in the Kaggle repository, https://www.kaggle.com/datasets/anlgrbz/student-demographics-online-education-dataoulad and

https://www.kaggle.com/datasets/anhtu96/ednet-kt34. The datasets were obtained from their respective publicly accessible sources and used in accordance with their applicable terms of use and data-access policies. The datasets support the learner interaction, assessment, activity, and knowledge-estimation analyses reported in this study. The data used for the experiments are available from the corresponding public dataset repositories.

DECLARATIONS

Ethics Approval and Consent to Participate: This study used publicly available educational learning analytics datasets, namely the Open University Learning Analytics Dataset (OULAD) and EdNet. No participants were directly recruited or contacted by the authors, and no new personally identifiable information was collected for this study. The analyses were conducted on the datasets as provided by their respective sources and in accordance with the applicable terms of use and data-access requirements. Therefore, no additional participant recruitment or consent procedures were undertaken by the authors.

REFERENCES

[1] Reddig, J.M., Arora, A., & MacLellan, C.J. (2025). Generating In-Context, Personalized Feedback for Intelligent Tutors with Large Language Models. *International Journal of Artificial Intelligence in Education, 35*, 3459 - 3500. https://doi.org/10.1007/s40593-025-00505-6

[2] Ahmed, Z., Shanto, S.S., Rime, M.H., Morol, M.K., Fahad, N., Hossen, M., & Abdullah-Al-Jubair, M. (2024). The Generative AI Landscape in Education: Mapping the Terrain of Opportunities, Challenges, and Student Perception. *IEEE Access, 12*, 147023-147050. https://doi.org/10.1109/ACCESS.2024.3461874

[3] Dahal, C., Chen, J., Ye, M., & Xiong, Z. (2026). Multi-Agent Systems in Education: A Survey from the Trustworthiness Perspective. *IEEE Transactions on Artificial Intelligence*. https://doi.org/10.36227/techrxiv.176704820.01980102/v1

[4] Youssef Alyoussef, I., Mohammed Drwish, A., Adel Albakheet, F., Hassan Alhajhoj, R., & Ahmed Al-Mousa, A. (2025). AI Adoption for Collaboration: Factors Influencing Inclusive Learning Adoption in Higher Education. *IEEE Access, 13*, 81690-81713. https://doi.org/10.1109/ACCESS.2025.3567656

[5] Algerafi, M.A., Zhou, Y., Alfadda, H., & Wijaya, T.T. (2023). Understanding the Factors Influencing Higher Education Students' Intention to Adopt Artificial Intelligence-Based Robots. *IEEE Access, 11*, 99752-99764. https://doi.org/10.1109/ACCESS.2023.3314499

[6] Oh, S. (2025). Integration of MATH41 and Generative AI in Pre-Service Mathematics Teacher Education: An Empirical Study on Lesson Design Competency. *IEEE Access, 13*, 128959-128973. https://doi.org/10.1109/ACCESS.2025.3586593

[7] Pradeesh, N., Thushara, M.G., Arun Krishna, K., Pranav, V., & Krishnamoorthy, S. (2025). AI-Driven Personalized Learning and Remedial Recommendation Through Knowledge Concept-Centric Evaluation. *IEEE Access, 13*, 207817-207837. https://doi.org/10.1109/ACCESS.2025.3638427

[8] Ma, Y., Tang, X., & Huang, X. (2025). AI-Powered Adaptive English Language Learning Systems: Leveraging Deep Learning Algorithms and Natural Language Processing for Personalized Teaching Approaches. *IEEE Access, 13*, 153189-153198. https://doi.org/10.1109/ACCESS.2025.3603602

[9] Lou, H., Yue, P., & Chen, J. (2025). Research on Adaptive Education Path Dynamic Programming Algorithm Based on Reinforcement Learning and Cognitive Graphs. *IEEE Access, 13*, 105446-105462. https://doi.org/10.1109/ACCESS.2025.3578389

[10] Viswanathan, N., Meacham, S., & Adedoyin, F.F. (2022). Enhancement of online education system by using a multi-agent approach. *Comput. Educ. Artif. Intell., 3*, 100057. https://doi.org/10.1016/j.caeai.2022.100057

[11] Chang, C., Shih, S., & Kuo, B. (2026). A Dual-Agent Intelligent Tutoring System for Teaching Two-Step Fraction Word Problems. *IEEE Transactions on Learning Technologies, 19*, 636-647. https://doi.org/10.1109/TLT.2026.3708780

[12] Rida, Z., Boukachour, H., Ennaji, M., & Machkour, M. (2025). Dynamic Adaptation in an Intelligent Multi-Tutoring System: A Multi-Agent Approach. *IEEE Access, 13*, 124587-124601. . https://doi.org/10.1109/ACCESS.2025.3576651

[13] Evangelista, E., & Salman Bukhari, S.M. (2026). From ethical principles to executable governance: A policy-as-code framework for trustworthy AI in higher education. *Computers and Education: Artificial Intelligence*. https://doi.org/10.1016/j.caeai.2026.100594

[14] Raji, N.R., Kumar, R.M., & Biji, C.L. (2024). Explainable Machine Learning Prediction for the Academic Performance of Deaf Scholars. *IEEE Access, 12*, 23595-23612. https://doi.org/10.1109/ACCESS.2024.3363634

[15] Lu, Y., Wang, D., Chen, P., & Zhang, Z. (2024). Design and Evaluation of Trustworthy Knowledge Tracing Model for an Intelligent Tutoring System. *IEEE Transactions on Learning Technologies, 17*, 1701-1716. https://doi.org/10.1109/TLT.2024.3403135

[16] Radharani, S., Sistla, V., & Kishore Kolli, V.K. (2026). CSBAM-MobileNet: a lightweight attention-enhanced deep learning model for student attentiveness classification. *Engineering Science and Technology, an International Journal*. https://doi.org/10.1016/j.jestch.2026.102326

[17] Yu, X., & Tian, Y. (2025). Enhancing Computer Education through IoT-Enabled Learning Environments Leveraging Mobile Edge Computing for Real-Time Feedback. *Systems and Soft Computing*. https://doi.org/10.1016/j.sasc.2025.200433

[18] Qorich, M., & Ouazzani, R.E. (2026). Context-based automatic question generation for educational assessment via lightweight transformer augmentation. *Eng. Appl. Artif. Intell., 177*, 114953. https://doi.org/10.1016/j.engappai.2026.114953

[19] Kalwar, A., Shekhawat, D., Joshi, S., & Bairwa, A.K. (2025). A Genetic Algorithm Approach to Adaptive Learning: Enhancing Learner Satisfaction and Performance. *IEEE Access, 13*, 102815-102829. https://doi.org/10.1109/ACCESS.2025.3578867

[20] Yuhana, U.L., Djunaidy, A. and Purnomo, M.H., (2024). Development of dynamic personalized learning paths based on knowledge preferences and the ant colony algorithm. IEEE Access, 12, pp.144193-144207. https://doi.org/10.1109/ACCESS.2024.3442312

[21] Zeng, Y., & Xu, X. (2022). The Construction and Optimization of an AI Education Evaluation Indicator Based on Intelligent Algorithms. *Int. J. Cogn. Informatics Nat. Intell., 16*, 1-22. https://doi.org/10.4018/IJCINI.315275

[22] Prasad G, Arun, et al. "Artificial Intelligence in Computer Science: An Overview of Current Trends and Future Directions." *Advances in Artificial and Human Intelligence in the Modern Era,* edited by S. Suman Rajest, et al., IGI Global Scientific Publishing, 2023, pp. 43-60. https://doi.org/10.4018/979-8-3693-1301-5.ch002

[23] Hesham Mohammed Ali Abdullah, Elham Alzain. AI-driven Predictive Analytics for Early Identification of At-risk StudentsAcademically in Saudi Universities., International Journal of Intelligent Engineering and Systems, Vol.19, No.6, 2026 DOI: 10.22266/ijies2026.0630.05

[24] Mosleh, Mohammed A. S. , et al. Adaptive Mirage Search Optimization for Educational Workload Scheduling in Multi-Cloud Environments. International Journal of Advanced Computer Science & Applications, 2026, v. 17, n. 7, p. 240

Data Analytics Framework for Crop Yield Prediction Based on the Impact of Climate Using a Hybrid Model

Narinder Kaur[1]*, Vishal Gupta[2]
NSUT East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research), USICT, Guru Gobind Singh Indraprastha University, Delhi, India[1]
NSUT East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research), New Delhi, India[2]

***Abstract*—In the modern Indian economy, agriculture is the most vital sector. Agriculture is an important source of revenue for the majority of the population of the world. Proper management of crops, crop species, crop field conditions, livestock, etc., can improve agricultural systems. The four primary subcategories of crop management are crop yield production management, weed management, crop disease management, and crop quality management. Planning agriculture, utilizing resources, and satisfying demand to ensure food security depend on accurate crop yield estimation. Crop yields are heavily influenced by climatic features like solar radiation, wind speed, rainfall, temperature, etc. Traditionally, crop yield management was completely performed by the farmers based on their experience using various techniques like crop rotation, mixed cropping, timely growing of crops, etc. These methods are combined with modern technologies like Artificial Intelligence, Internet of Things, drones, precision agriculture, etc., to improve crop productivity while considering preservation of natural resources. This study performed crop yield prediction using four baseline models based on machine learning: Decision Tree, Multiple Linear Regression, Support Vector Regression, and Random Forest. One of the best baseline models is further improved using nature-inspired algorithms. Four nature-inspired optimization algorithms used are: Whale Optimization Algorithm, Harris Hawks Optimization, Particle Swarm Optimization, and Grey Wolf Optimization. Particle Swarm Optimization-based optimized RF performed the best with R^2=98.32, MAE=9.2766.**

***Keywords*—Crop yield prediction; impact of climate change; decision tree; support vector regression; linear regression and random forest; particle swarm optimization; whale optimization algorithm; Harris Hawks optimization; grey wolf optimization**

I. Introduction

Climate change is an important challenge that impacts various living beings and our environment directly or indirectly. Continuous change in climate and the rise of CO2 concentration in our environment are making agricultural system enhancement more challenging [1]. The Food and Agriculture Organization (FAO) reports that there has been a sharp rise in grain consumption and demand as compared to crop production. From 1964 to 2030, the demand for rice, wheat, and other coarse grains increased steadily [2].

In agriculture, food security is highly threatened by climate change, as this change affects crop production. The impact of climate change on agriculture includes sudden increases or decreases in temperature, variable precipitation patterns, and a rise in the frequency of weather events [3][4]. In the agricultural sector, weather forecasting affects when crops are planted, harvested, and irrigated. This eventually helps to maximize crop yields and ensures food security [5]. Estimating crop production is a crucial component that depends on a number of variables, like soil characteristics, climate, and irrigation [6].

In the traditional crop assessment system, assessing soil health, crop growth, crop yield prediction, crop field management, etc. was done through manual observations, agricultural surveys, or by agricultural experts. However, wrong assessment led to reduced crop production. Numerous mitigating techniques have been employed over the past few decades, but the environment is still changing. Predicting crop yield is a challenge in modern precision agriculture, despite the proposal of numerous efficient models [7]. These models depict how temperature and precipitation variations directly impact crop production due to climate change.

The improvement of agricultural systems is greatly aided by Machine Learning (ML) approaches. ML is a scientific field that enables machines to learn without explicit programming [8]. Agriculture is increasingly using machine learning techniques to track and forecast environmental variables [9]. Numerous ML-based models have been proposed for agricultural system improvement [10]. However, managing massive volumes of data for precise conclusions and identifying the most significant climatic characteristics are the primary areas of worry. This can be resolved by employing an effective method that can also improve forecast performance.

In order to create more intelligent, quicker, and flexible systems, software engineers are increasingly using these natural designs. We refer to this method as "nature-inspired algorithms." It entails researching how organisms act, such as how ants locate food, how the brain learns, and how ecosystems maintain equilibrium, and then transforming those behaviours into functional software solutions. In order to solve open-ended problems, nature-inspired optimization strategies are both necessary and beneficial. Researchers and academicians can pursue a variety of avenues by comprehending nature-inspired methods and their significance for resolving practical issues.

A model for predicting agricultural crop yield while taking weather conditions into account has been proposed in this

*Corresponding author.

research. The following is the planned work's primary contribution:

- A complete data-analytic-based framework is created using existing algorithms for prediction of crop yield based on the impact of climate change through nature-inspired optimizing algorithms. The framework establishes how climate and agricultural variables can be incorporated into crop-yield prediction.
- Feature selection is performed using the RF approach.
- Crop yield prediction is performed using Multiple Linear Regression (MLR), Decision Tree (DT), Support Vector Regression (SVR), and Random Forest (RF) baseline models.
- The RF model is optimized using Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Harris Hawks Optimization (HHO), and Grey Wolf Optimization (GWO) to predict crop yield.

This study's remaining sections are arranged as follows: Section II illustrates literature related to the proposed model. The suggested Crop Yield Prediction technique is briefly described in Section III. Section IV illustrates evaluation metrics used for comparison. Section V illustrates results and discussion. In Section VI, the entire study project is finally concluded.

II. Related Works

Numerous studies have been done in this field, which include identifying the best crop(s) and predicting crop yields using weather and soil information. Different articles related to the proposed work are summarized as follows:

Ramesh et al. [11] performed crop yield prediction using Neural Network (NN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Deep Neural Network (DNN) and proposed a hybrid model that combines DNN with WOA and Adaptive PSO, which performed better than the other models. Mohanty et al. [12] performed rice yield forecasting at the district level in Odisha using MLR and Quantile Regression (QR) as base models and proposed a hybrid model that merges RF, eXtreme Gradient Boosting (XGBoost), and Artificial Neural Networks (ANN) models based on crop and climate data of Odisha. Chaurasia et al. [13] proposed a hybrid model that combines LR, Support Vector Machines (SVM), k-Nearest Neighbors (kNNs), DT, RF, and XGBoost based on ensemble learning for crop yield forecasting.

Manjunath et al. [14] performed crop yield prediction using Random Forest (RF), XGBoost, Decision Tree (DT), SVM, and Linear Regression (LR) and then merged the three best of these models using the ensemble learning (Voting) technique and concluded that the hybrid model performed better. Dehghanisanij et al. [15] proposed a hybrid model that uses Adaptive Neuro-Fuzzy Inference System (ANFIS) and the Seasons Optimization (SO) algorithm to estimate water-use efficiency and yield in apple orchards. Cao et al. [16] performed wheat yield prediction at country level and field level using traditional models: RF and three deep learning-based models: Deep Neural Networks (DNN), 1D Convolutional Neural Networks (1D-CNN), and Long Short-Term Memory (LSTM) model, and concluded that all four models performed well at country-level prediction, whereas DNN and RF performed too well at field-level prediction.

Khaki et al. [17] performed crop yield prediction of corn and soybean based on archived yield data, climate data, and soil data using a hybrid model that combines Convolutional Neural Network and Recurrent Neural Network (CNN-RNN). Han et al. [18] performed wheat yield prediction based on Climate, soil, vegetation index, cropland, and crop yield data using SVM, Gaussian process regression (GPR), RF, DT, NN, k-NN, boosted trees, and bagging trees, and concluded that SVM, RF, and GPR performed better than the other models. Sun et al. [19] proposed a hybrid model, CNN-LSTM, for prediction of Soybean yield based on past yield data, climate data, and Moderate Resolution Imaging Spectroradiometer (MODIS) data. Gopal et al. [20] proposed a hybrid model, MLR-ANN, to predict paddy yield based on Cropland, climate, soil, and past yield data. Hilal et al. [21] performed yield prediction of palm oil based on past yield data, cropland data, climatic data, and air pollutants data using Genetic algorithm/correlation analysis (GA/CA).

Sánchez et al. [22] performed crop yield prediction of multiple crops using Multiple linear regression, M5-Prime regression trees, perceptron multilayer neural networks, and concluded that M5-Prime and k-nearest neighbors performed better than other algorithms. To effectively predict agricultural productivity, it is crucial to observe climate patterns during sowing, the growing season, and throughout the year [23]. It is very difficult to accurately anticipate the crop growth stage using climate data because of very complicated dynamics that result in considerable changes in climatic factors [24]. Deep models used in [11], [17], [19], etc. have hidden parameter interactions. Stacking or voting architectures used in [12], [13], [14], etc. combine powerful baseline models but leave their hyperparameters fixed or unoptimized. In order to mitigate these concerns, this study has proposed an effective crop yield prediction model using RF optimization.

III. Methodology

Crop yield prediction while considering climate change is one of the important challenges in agriculture as climatic variables are fluctuating with respect to time. We depend on it to ensure food security and fulfill the future's food demand so as to accommodate the increase in the world's population of our world.

Fig. 1 depicts the methodology used for crop yield prediction. The proposed model is suitable for better prediction of yield, which can be used to increase crop yield production to fulfil future food demand. The model takes data related to crops and climate as an input. The input dataset is preprocessed by removing irrelevant columns and duplicate values. Then it is split into training data and Test Data. After splitting, One-Hot Encoding and normalization are performed in order to improve the classification accuracy. Following that RF based features selection is performed. The next step is to train the model using four machine learning baseline models: DT, MLR, SVR, RF and crop yield prediction is performed. Out of these four baseline

models, the best model is selected for further improvement using nature-inspired algorithms. Four RF optimized models: PSO optimized RF; WOA optimized RF; HHO optimized RF; and GWO optimized RF are trained and crop yield prediction is performed. The proposed model is explained as follows:

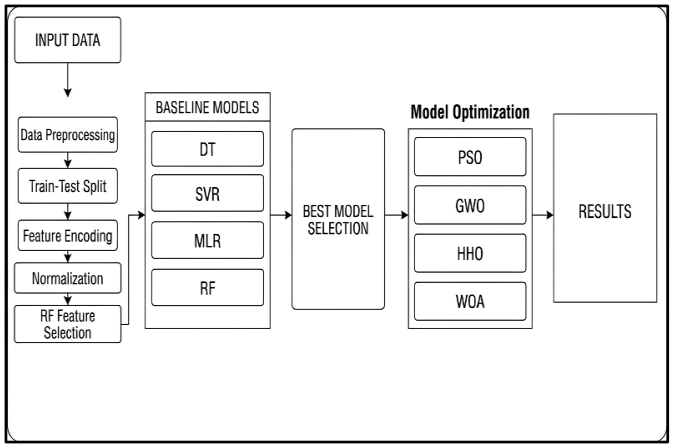


Fig. 1. The proposed model for crop yield prediction.

A. Crop and Climate Data

The crop yield dataset includes crop and climate data of different states of India taken from www.kaggle.com [36]. Dataset includes 19,689 unique samples/records. The dataset covers all states in India. The prediction based on the dataset can be used for improvement in the current agriculture system.

B. Data Preprocessing

In this part of the model, stateless preprocessing operations like removal of irrelevant columns and duplicate values, exploring dataset, separating the target variable from the others, etc., are done in this phase.

C. Train-Test Split

In the proposed model, two-way splitting is performed, splitting the data into training data and Test data. The model is fed the Train data in order to identify patterns and establish its own internal rules. The Test data serves as a last check to verify if the model can accurately predict data that it has never seen. In ML, to evaluate model performance, a Train-Test split is utilized. This split also avoids overfitting and helps in real-world simulation.

D. One-Hot Encoding and Normalization

A data preprocessing method called "One-Hot Encoding" is employed to convert categorical data into a numerical representation that is understandable by ML algorithms. For every category, it generates distinct binary columns, where 0 denotes the absence of a category, and 1 denotes its existence. The next step is to convert the encoded data into a form that is required by the data mining algorithms and improves the model. This conversion or transformation is known as data normalization. Here, Z-score normalization is used for data normalization [Eq. (1)].

$$Z = \frac{X-\mu}{\sigma} \tag{1}$$

where, X represents the actual value, μ represents the population's mean, and σ is the population's deviation. A correlation heat map is used for initial feature selection.

E. RF-Based Feature Selection

Feature selection is a process to select features that are most important for the model. By lowering overfitting, increasing accuracy, and cutting down on calculation time, it may improve the performance of the model. Feature selection, which involves figuring out those features which are most relevant for building a predictive model, is a crucial step in the machine learning process.

We conducted an ablation study by performing features selection using PSO, WOA, GWO, HHO, BACO (Binary Ant Colony Optimization) and RF. But PSO, WOA, HHO, GWO, and BACO selected features that excluded the main feature; no climate features were selected by these algorithms. Therefore, we used RF feature selection for the same, as it selected climate features, which justifies this manuscript.

F. Crop Yield Prediction

Crop Yield Prediction is performed using the features selected by RF through eight models: four baseline models DT, SVR, MLR, RF, and four optimized RF models, which are optimized using PSO, GWO, HHO, and WOA optimization algorithms. The following part of this section briefly explains these models. Training dataset with N samples and p features is expressible as:

$$Dataset = \{(X_i, Y_i)\}_{i=1}^{N} \tag{2}$$

where, $X_i = (X_{i1}, X_{i2}, \dots, X_{ik})$ i.e. ith sample of the $Dataset$, Y_i is the original yield of ith sample, N is the total count of samples, $k = 1,2, \dots, p$ is a feature's index.

G. Decision Tree (DT)

A supervised learning-based approach used in ML, data mining, and statistics is decision tree learning. The DT approach can be used for classification as well as for regression. Here, it is used for regression purposes. DT approach creates a hierarchical tree structure that includes a single base node, branches, internal nodes, and leaf nodes. It functions similarly to a flowchart for step-by-step decision making [25] [26]. Fig. 2 depicts the Decision Tree Regression Approach briefly.

In DT, firstly, the mean of the node is computed as:

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{3}$$

where, $\bar{Y}$ represents the mean crop yield of the node, n represents total samples' count in the node, and Y_i is the actual crop yield of the ith sample in that node. After it Mean Squared Error is computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2 \tag{4}$$

The next step is to split, every feature is tested for different threshold values, and the weighted MSE is computed as:

$$MSE_{split} = \frac{|S_{Left}|}{|S|} MSE_{Left} + \frac{|S_{Right}|}{|S|} MSE_{Right} \tag{5}$$

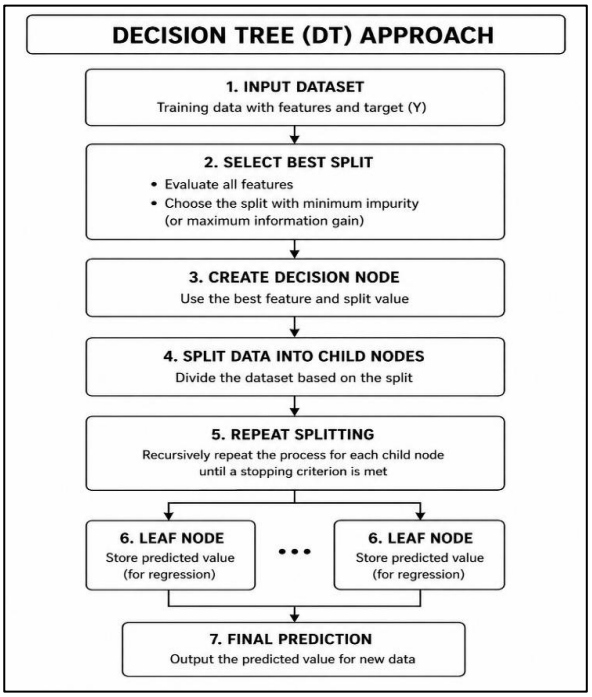


Fig. 2. Decision Tree (DT) working.

where, S is count of samples before splitting, S_{Left} is the count of samples in the left split, S_{Right} is count of samples in the right split, MSE_{Left} is the left split's Mean Square error, and MSE_{Right} is the right split's Mean Squared Error. MSE_{Left} *and* MSE_{Right} can be computed as:

$$MSE_{Left} = \frac{1}{S_{Left}} \sum \left(Y_{i\_Left} - \bar{Y}_{Left}\right) \quad (6)$$

$$MSE_{Right} = \frac{1}{S_{Right}} \sum \left(Y_{i\_Right} - \bar{Y}_{Right}\right) \quad (7)$$

where, Y_{i_Left} is the actual crop yield of a sample in the left node and Y_{i_Right} is the actual crop yield of a sample in the right node, $\bar{Y}_{Left}$ is mean crop yield of the left node, $\bar{Y}_{Right}$ is mean of crop yield of the right node. The best split is selected as:

$$(k^*, t^*) = arg\ arg\ MSE_{split}\ (k, t) \quad (8)$$

where, k is candidate's feature index, t is the candidate's threshold value, k^* is the best feature which is selected for split, t^* is the best feature's threshold, $MSE_{split}(k, t)$ is weighted MSE computed using j feature and t threshold value. Splitting is done till we reach a leaf node. Once a new sample reaches a leaf node, compute prediction as:

$$\hat{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i \quad (9)$$

where, $\hat{Y}$ is the predicted crop yield of the node, n represents total samples' count in the leaf node, Y_i actual crop yield of each sample in the leaf.

The reduction of the Mean Squared Error (MSE) is the primary goal of DT:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(Y_i - \hat{Y}_i\right)^2 \quad (10)$$

where, N is the training sample count, Y_i is actual crop yield of ith sample, $\hat{Y}_i$ is the predicted crop yield of the ith sample, $\left(Y_i - \hat{Y}_i\right)^2$ is the residual error squared.

H. Multiple Linear Regression (MLR)

MLR Targets Defining a Linear Relationship between a Dependent Variable and Numerous Independent Variables. Researchers Can Investigate the Effect of Multiple Independent Variables on the response simultaneously by Using Multiple Regression [27]. Fig. 3 depicts the working of the MLR Model.

MLR equation for crop yield of a sample is as follows:

$$Y_i = f(X_{i1}, X_{i2}, \ldots, X_{ik}) \quad (11)$$

Eq. (11) can be converted to the following form:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik} + \epsilon_i \quad (12)$$

where, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients of features $(X_1, X_2, \ldots, X_k)$ and ϵ_i represents the random error term of ith sample $i = 1, 2, \ldots, N$.

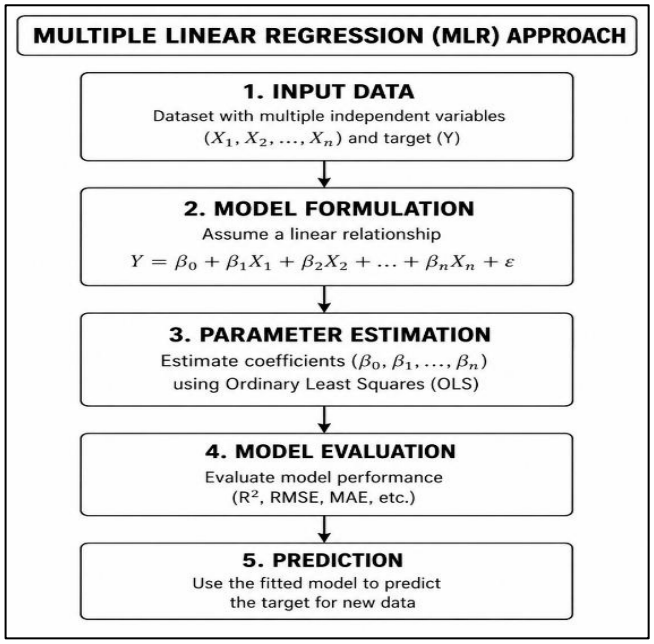


Fig. 3. Multiple Linear Regression (MLR) working.

The complete matrix representation of the above Eq. (12) can be written as:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1k} \\ 1 & X_{21} & \cdots & X_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & \cdots & X_{nk} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad (13)$$

Ordinary Least Squares (OLS) solution is used to compute the regression coefficients as:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (14)$$

MLR equation for crop yield prediction of a sample is as follows:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \hat{\beta}_2 X_{i2} + \cdots + \hat{\beta}_n X_{ik} \quad (15)$$

Residual error (e) is computed as:

$$e = Y_i - \hat{Y}_i \quad (16)$$

The reduction of the Mean Squared Error (MSE) is the primary goal of MLR also [same as of DT, Eq. (10)].

I. Support Vector Regression (SVR)

In 1996, Vladimir N. Vapnik, Harris Drucker, Christopher J. C. Burges, Linda Kaufman, and Alexander J. Smola introduced a regression using SVM. We refer to this technique as support vector regression (SVR). SVR aims at finding the optimal fit that precisely predicts the objective variable while lowering complexity to prevent overfitting. This is accomplished by defining an ε-insensitive tube around the hyperplane that allows for a certain amount of variance between the actual and anticipated target values. A machine learning method called non-linear Support Vector Regression (SVR) is used to forecast continuous values in situations where data associations are complicated or curved [28]. Fig. 4 represents the working of the SVR approach.

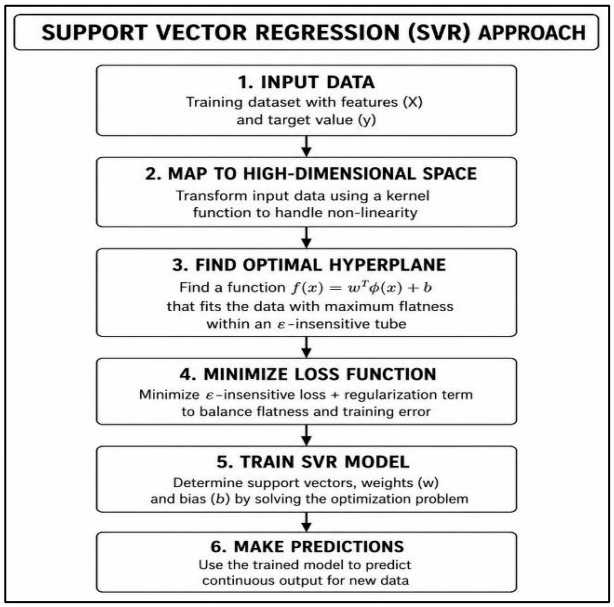


Fig. 4. Support Vector Regression (SVR) working.

The SVR non-linear function for prediction:

$$f(X) = \hat{Y} = W^T\phi(X) + \gamma \quad (17)$$

Where W^T represents the transpose of weight vector W, $\phi(X)$ is nonlinear mapping to a higher-dimensional feature space, γ is the bias.

SVR ignores small errors, known as error-insensitive, which can be written as:

$$L_\epsilon\big(Y, f(X)\big) = \begin{cases} 0 & |Y - f(X)| \le \epsilon \\ |Y - f(X)| - \epsilon & |Y - f(X)| > \epsilon \end{cases} \quad (18)$$

Where, ϵ is the width of the error-insensitive tube. SVR objective is to optimize (minimize):

$$min\,\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (19)$$

with the following conditions:

$$Y - W^T\phi(X_i) - b \le \epsilon + \xi_i \quad (20)$$

$$W^T\phi(X_i) + b - Y_i \le \epsilon + \xi_i^* \quad (21)$$

$$\xi_i, \xi_i^* \ge 0 \quad (22)$$

Where, $\|W\|^2$ is Euclidean norm of W, C is the regularization parameter, ξ_i is error above the boundary, and ξ_i^* is error below the boundary.

Eq. (19) can be converted to a dual optimization problem using Lagrange multipliers as:

$$max\,-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(X_i, X_j) - \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}Y_i(\alpha_j - \alpha_j^*) \quad (23)$$

with conditions:

$$0 \le \alpha_i, \alpha_i^* \le C \quad (24)$$

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \quad (25)$$

where, i is first sample index, j is the second sample index, α_i is Lagrange multiplier which is associated with upper error-insensitive constraint for ith sample, α_i^* is Lagrange multiplier which is associated with the lower error-insensitive constraint for ith sample, $K(X_i, X_j)$ is the function of kernel that measures the similarity between the ith and jth samples in the transformed feature space, X_i is ith training sample, X_j is jth training sample, ϵ defines the width of the insensitive tube, Y_i is the target value, C is the regularization parameter.

Kernel function used in equation is expressed as:

$$K\big(X_i, X_j\big) = exp\left(-\Upsilon\sum_{k=1}^{p}\big(x_{ik} - x_{jk}\big)^2\right) \quad (26)$$

The final prediction equation of SVR is as follows:

$$f(X) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)exp(-\Upsilon\|X_i - X\|^2) + b \quad (27)$$

J. Random Forest (RF) Regression

RF approach is an ensemble learning approach that creates numerous decision trees and uses bagging to mix their outputs to prevent overfitting and improve prediction accuracy. Bagging and boosting ensemble learning are the two primary categories of ensemble learning techniques. In bagging ensemble learning, distinct sub-datasets are generated from the primary dataset and sent to various machine learning models. The final output or outputs are created by combining the findings of these simultaneous models. In boosting, several machine learning models use the same dataset. These models operate in a sequential manner. The first model's findings are improved by the second model, the second model's findings are improved by the third model, and so on. Bagging is the foundation of the RF method. The Random Forest automatic machine learning technique was developed in 2001 by Leo Breiman and Adèle Cutler. [29]. Fig. 5. depicts working of the RF model.

In RF regression, Multiple trees T are used in RF approach, and for each tree, a separate Dataset D_t is created. Creating of different datasets is called bootstrap sampling and can be written as:

$$D_t \subseteq D, for\ t = 1,2, \dots, T \quad (28)$$

where, D_t is bootstrap dataset for the t[th] tree, and T stands for total count of trees.

Each tree predicts the output i.e., crop yield independently as:

$$\hat{Y}_t = f_t(X) \quad (29)$$

where, $\hat{Y}_t$ is the prediction of t[th] tree, X is input feature vector.

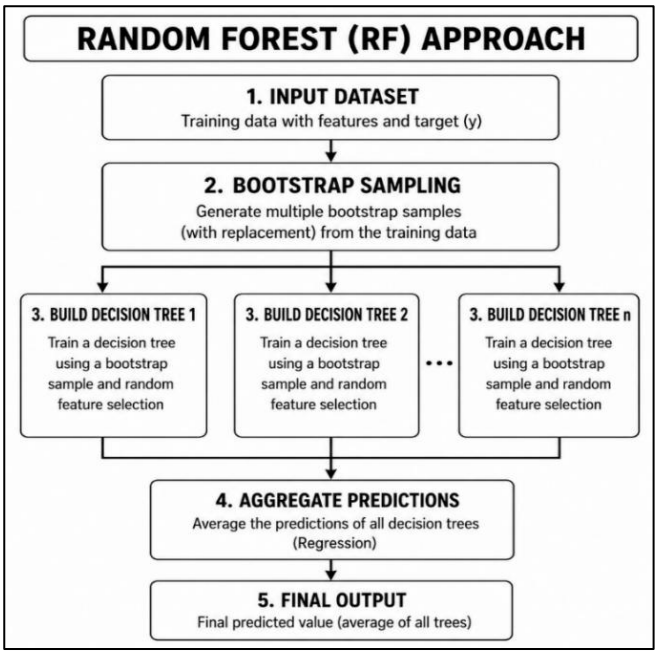


Fig. 5. Random Forest (RF) working.

The final RF Prediction equation can be written as:

$$\hat{Y} = \frac{1}{T}\sum_{i=1}^{T} f_t(X) \quad (30)$$

where, $\hat{Y}$ final predicted crop yield, T stands for the count of trees, $f_t(X)$ is predicted value of the t[th] tree.

K. Optimized RF Models

Here, four nature-inspired optimization approaches are used to optimize the RF model: PSO, GWO, HHO, and WOA algorithms. These optimization algorithms find the best parameters for the RF model.

Let d be the number of parameters of RF model to be optimized; then initialize the Lower Bound (LowB) and Upper Bound (UpB) vectors to specify the lower and upper limits of each RF parameter that is to be optimized as:

$$LowB = [LowB_1, LowB_2, \dots LowB_d] \quad (31)$$

$$UpB = [UpB_1, UpB_2, \dots UpB_d] \quad (32)$$

Initialization of position matrix X is as follows:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_{11} & \cdots & X_{1d} \\ X & \cdots & X_{2d} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nd} \end{bmatrix} \quad (33)$$

where, count of rows= n represents no. of swarms (in PSO)/ wolves (in GWO)/ hawks (in HHO)/ whales (in WOA), count of columns= d represents parameters of the RF model to be optimized. A single row represents a single particle (in PSO)/ wolf (in GWO)/ hawk (in HHO)/ whale (in WOA) as $X_i = [X_{i1}, X_{i2}, \dots, X_{id}]$. Initialization of every value in the above matrix is done as follows:

$$X_{ij} = LowB_j + r_{ij}(UpB_j - LowB_j) \quad (34)$$

where, X_{ij} is the single value of the matrix shown in equation, UpB_j is the Upper Bound value of parameter j, $LowB_j$ is the Lower Bound value of parameter j, r_{ij} is a random variable such that $0 \leq r_{ij} \leq 1$.

Eq. (31), (32), (33), and (34) are common in all four algorithms and belongs to the initialization part of it.

These four optimized RF models compute the fitness of a single row as:

$$Fitness(X_i) = RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{Y}_i)^2} \quad (35)$$

L. PSO Optimized RF

PSO was first identified in 1995 by Kennedy and Eberhart. PSO technique is used to optimize continuous nonlinear functions. PSO is a stochastic population-based optimization method inspired by natural swarm intelligence. Every particle travels across the search space and represents a possible solution. Both individual experience and collective swarm knowledge direct movement. The approach uses a fitness function to iteratively improve solutions [30][31]. Fig. 6 represents the working of PSO-based RF optimization.

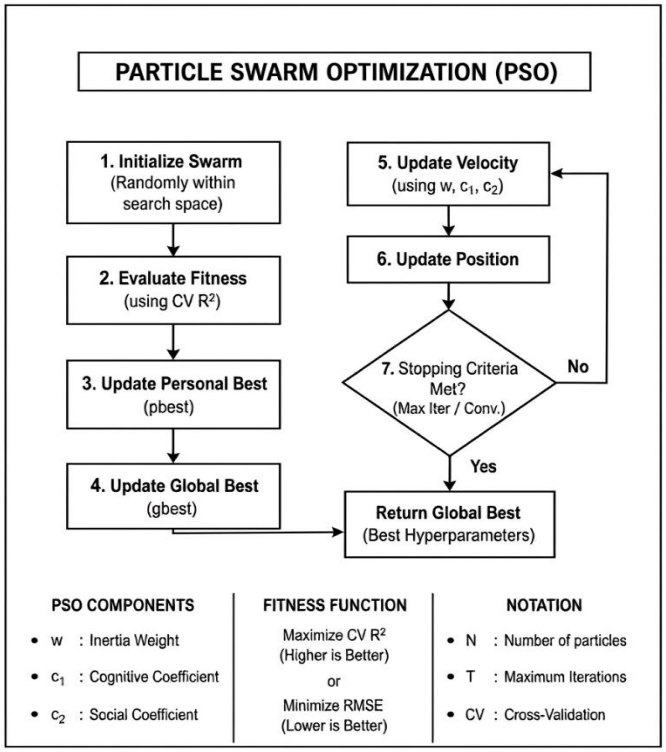


Fig. 6. Particle Swarm Optimization (PSO) working.

Steps to optimize RF model using PSO:

Initialization:

Initialize Lower Bound (LowB) and Upper Bound (UpB) vector: using Eq. (31) and (32)

Initialization of the swarm position matrix: using Eq. (33) and assign values to it using Eq. (34)

Initialize the velocity matrix

$$V = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} V_{11} \cdots V_{1d} \\ V_{21} \cdots V_{2d} \\ \vdots \quad \ddots \quad \vdots \\ V_{n1} \cdots V_{nd} \end{bmatrix} \quad (36)$$

Compute Fitness of every particle using Eq. (35)

Personal best is updated as:

$$PB_i = \begin{cases} X_{i,} & if\ Fitness(X_i) > Fitness(PB_i) \\ PB_{i,} & otherwise \end{cases} \quad (37)$$

where, PB_i best solution found by particle i

Global best's update as:

$$GB = \arg\max_i x\, Fitness(P_i) \quad (38)$$

where, GB is the best solution found by entire swarm.

Update Velocity and particle position:

Velocity update:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (PB_i - X_i^t) + c_2 r_2 (GB - X_i^t) \quad (39)$$

where, ω represents inertia weight, c_1 is cognitive coefficient, c_2 is the social coefficient, t represents current iteration.

Particle position update:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (40)$$

Repeat steps 2 to 5 until convergence is attained or the maximum number of iterations is attained.

RF is optimized with the best solution as,

$$X_{best} = GB \quad (41)$$

M. GWO Optimized RF

A nature-inspired metaheuristic algorithm called Grey Wolf Optimization (GWO) that imitates the leadership organization and hunting techniques of grey wolves in the wild. Seyedali Mirjalili first presented it in 2014. The best fit, according to GWO, is the Alpha wolf (α); the second best fit is the Beta wolf (β),the third best fit is the Delta wolf (δ); and the Omega wolves (ω) are the remaining possible options [32][33]. Fig. 7 depicts the working of GWO Based RF optimization.

Steps to optimize the RF model using GWO:

Initialization:

Initialize Lower Bound (LowB) and Upper Bound (UpB) vector: using Eq. (31) and (32)

Initialization of the wolf position matrix: using Eq. (33) and assign values to it using Eq. (34)

Compute Fitness for every wolf using Eq. (35)

Identify the Alpha (α), the Beta (β), the Delta (δ) and Omega (ω) wolves as:

$$X_\alpha = X_i\ with\ best\ Fitness(X_i) value \quad (42)$$

$$X_\beta = X_i\ \ with\ second\ best\ Fitness(X_i) value \quad (43)$$

$$X_\delta = X_i\ with\ third\ best\ Fitness(X_i) value \quad (44)$$

$$X_\omega = X_i\ with\ rest\ of\ the\ wolves \quad (45)$$

Compute parameter a as:

$$a = 2 - \frac{2t}{T} \quad (46)$$

where, t shows the current iteration, and T stands for maximum iteration. Now compute the coefficient as:

$$A = 2_{ar_1} - a \quad (47)$$

$$C = 2r_2 \quad (48)$$

where, A and C are coefficients, where r_1 and r_2 are random values such that $0 \le r1, r2 \le 1$

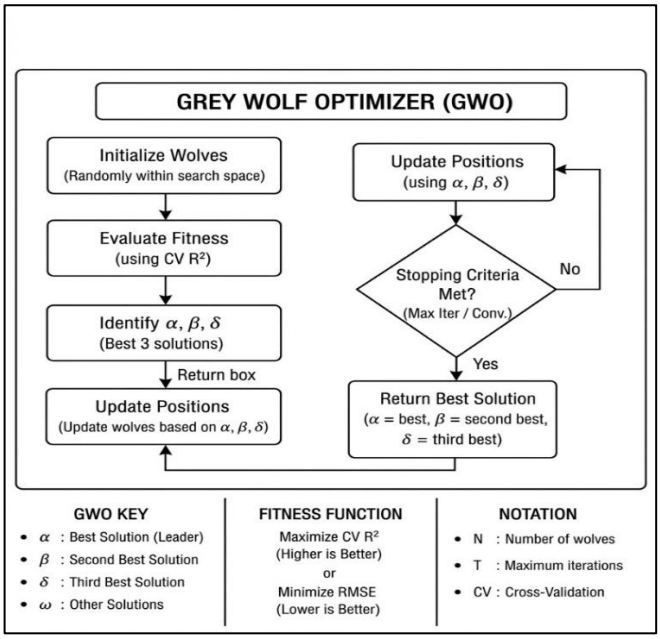


Fig. 7. Grey Wolf Optimization (GWO) working.

Compute the distance from X_α, X_β, and X_δ wolves and Compute candidate positions based on D_α, D_β, and D_δ as:

Distance from X_α, X_β, and X_δ wolves as:

$$Distance_\alpha = |C_1 X_\alpha - X| \quad (49)$$

$$Distance_\beta = |C_1 X_\beta - X| \quad (50)$$

$$Distance_\delta = |C_1 X_\delta - X| \quad (51)$$

Candidate positions based on D_α, D_β, and D_δ as:

$$X_1 = X_\alpha - A_1 Distance_\alpha \quad (52)$$

$$X_2 = X_\beta - A_2 Distance_\beta \quad (53)$$

$$X_3 = X_\delta - A_3 Distance_\delta \quad (54)$$

Update Wolf Position as:

$$X_{t+1} = \frac{X_1 + X_2 + X_3}{3} \quad (55)$$

Eq. (55) will update every wolf except Alpha wolf.

Check the updated position for its upper bounds and lower bounds as

$$min(UpB, max(LowB, X)) \quad (56)$$

Evaluate Fitness again and Update X_α, X_β, and X_δ again

Until the maximum count of iterations is attained, repeat steps 2 through 7.

RF is optimized with best solution as:

$$X_{best} = X_\alpha \quad (57)$$

N. HHO Optimized RF

Ali Asghar Heidari and associates introduced the well-known population-based, nature-inspired HHO metaheuristic algorithm in 2019. It imitates Harris's hawks' coordinated foraging and "surprise pounce" hunting techniques. Exploration, escaping energy shifts, and exploitation tactics are important elements of the HHO optimizer [34]. Fig. 8 represents the working of HHO-based RF optimization.

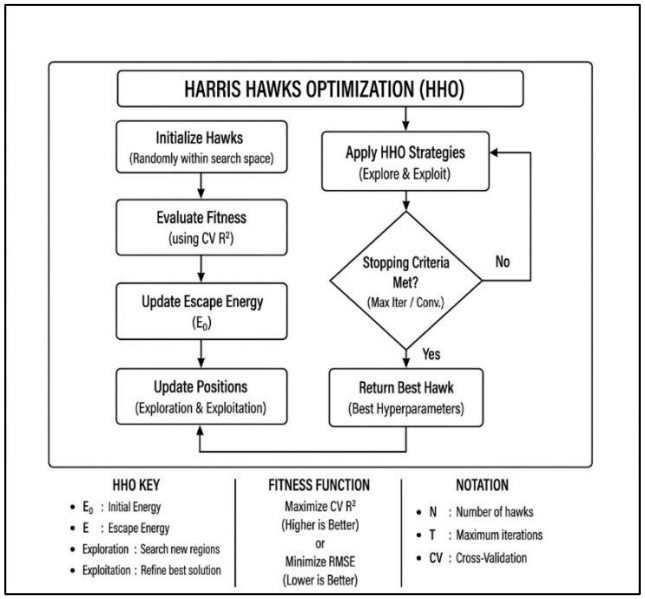


Fig. 8. Harris Hawks Optimization (HHO) working.

Steps to optimize RF model using HHO:

Initialization:

Initialize Lower Bound (LowB) and Upper Bound (UpB) vector: using Eq. (31) and (32)

Initialization of Hawks position matrix: using Eq. (33) and assign values to it using Eq. (34)

Compute Fitness for every hawk using Eq. (35)

Identify the best hawk (Rabbit) as:

$$X_{rabbit} = \arg\max_i Fitness(X_i) \quad (58)$$

Calculate the escaping energy that regulates the change from exploration to exploitation using the equation below:

$$E_0 = 2r - 1 \quad (59)$$

where, r is a random value such that $0 \le r \le 1$, current escaping energy is expressed as:

$$E = 2E_0\left(1 - \frac{t}{T}\right) \quad (60)$$

where, t shows the current iteration and T stands for the maximum number of iterations

Exploration phase: When $|E| \ge 1$, the hawk explores the search space as:

$$X(t+1) = X_{random}(t) - r1|X_{random}(t) - 2r_2X(t)| \quad (61)$$

where, X_{random} is a randomly selected hawk.

Explotation phase: When $|E| < 1$, hawk exploits the best solution (Rabbit) as :

$$X(t+1) = X_{rabbit} - E|JX_{rabbit} - X(t)| \quad (62)$$

where, $J = 2(1 - r)$, and r is a random such that $0 \le r \le 1$

Checking new computed value for boundaries using UB and LB as

$$X = min(UpB, max(LowB.X) \quad (63)$$

Evaluate Fitness again

Update the Rabbit again

Until the maximum number of iterations is attained, repeat steps 2 through 9.

RF is optimized with best solution as

$$X_{best} = X_{rabbit} \quad (64)$$

O. WOA Optimized RF

Seyedali Mirjalili and Andrew Lewis introduced the Whale Optimization method (WOA), a metaheuristic optimization method inspired by nature, in 2016. Only humpback whales exhibit the unusual behavior known as "bubble-net feeding." The whale optimization algorithm (WOA) provides a theoretical description of the spiral bubble-net feeding maneuver. WOA mimics hunting behavior by pursuing the prey with a random or optimum search agent. WOA simulates humpback whale bubble-net attacks using a spiral [35]. Fig. 9 depicts WOA-based RF optimization working.

Steps to optimize RF model using WOA:

Initialization:

Initialize Lower Bound (LowB) and Upper Bound (UpB) vector: using Eq. (31) and (32)

Initialization of Whale position matrix: using Eq. (33) and assign values to it using Eq. (34)

Compute Fitness for every whale using Eq. (35)

Identify best whale as:

$$X^* = \arg\max_i Fitness(X_i) \quad (65)$$

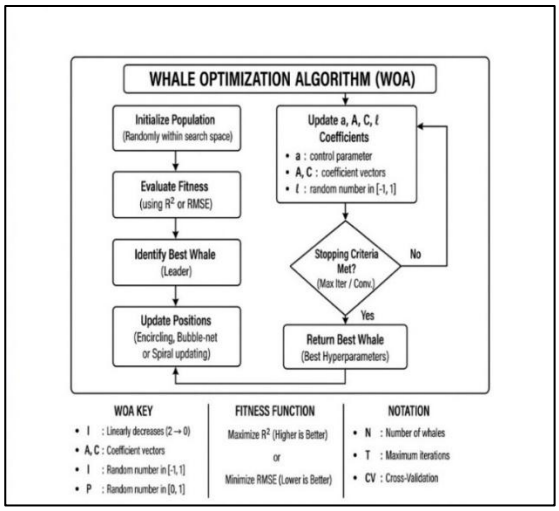


Fig. 9. Whale Optimization Algorithm (WOA) working.

Update control parameter as:

$$a = 2 - \frac{2t}{T} \quad (66)$$

where, t shows the current iteration and T stands for the maximum number of iterations

Compute Coefficient Vectors as:

$$A = 2_{ar_1} - a \quad (67)$$

$$C = 2r_2 \quad (68)$$

where, A and C are coefficients, where r_1 and r_2 are random values such that $0 \leq r1, r2 \leq 1$

Choose the search strategy based on a randomly generated p value between 0 and 1 as follows:

Case 1: If $p < 0.5$ and $|A| < 1$, compute distance to whale and update position as:

$$Distance = |CX^* - X| \quad (69)$$

$$X(t+1) = X^* - ADistance \quad (70)$$

Case 2: If $p < 0.5$ and $|A| \geq 1$, compute distance to whale and update position as:

$$D = |CX_{random} - X| \quad (71)$$

$$X(t+1) = X_{rand} - ADistance \quad (72)$$

Case 3: If ≥ 0.5 , compute distance to whale and update position as:

$$Distance' = |X^* - X| \quad (73)$$

$$X(t+1) = Distance' e^{bl} \cos(2\pi l) + X^* \quad (74)$$

where, b is the spiral constant, generally taken as 1, l ranges $-1 \geq l \geq 1$

Checking new computed value for boundaries using J and LB as

$$X = min(UpB, max(LowB.X)) \quad (75)$$

Evaluate Fitness again

Update the best whale as:

$$X^* = X_i \quad (76)$$

Repeat steps 2 to 9 until the maximum number of iterations are reached.

RF is optimized with best solution as

$$X_{best} = X^* \quad (77)$$

IV. Evaluation Metrics

The performance of the proposed crop yield prediction framework is examined using R Squared, Mean Absolute Error, Root Mean Square Error, and 5-Fold CV Score.

A. Mean Absolute Error (MAE)

The average absolute difference between actual and predicted data is measured as mean absolute error. It provides results in the target variable's unit. It can be evaluated as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \hat{Y}_i\right| \quad (78)$$

where, N count of observations is, Y_i is the actual value, $\hat{Y}_i$ is the predicted value.

B. Root Mean Square Error (RMSE)

The square root of Mean Squared Error (MSE) is called Root Mean Squared Error. It improves interpretability by returning the result to the target variable's original unit while maintaining the severe penalty for significant errors. It is evaluated as:

$$RMSE = \sqrt{MSE} \quad (79)$$

$$\text{where, } MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{Y}_i)^2 \quad (80)$$

C. R Squared (R^2)

It is also known as the Coefficient of Determination. R-squared indicates how much of the variance in the target variable can be explained by the regression model. It is judged as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y}_i)^2} \quad (81)$$

D. Cross-Validation Score (CV)

$$R_k^2 = 1 - \frac{\sum_{i=1}^{Nk}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{Nk}(Y_i - \bar{Y}_i)^2} \quad (82)$$

$$CV_{Score} = \frac{1}{k}\sum_{k=1}^{5} R_k^2 \quad (83)$$

5-fold CV Score is used in the proposed model as:

$$CV_{Score} = \frac{R_1^2 + R_2^2 + R_3^2 + R_4^2 + R_5^2}{5} \quad (84)$$

V. Results and Discussion

All four of the optimized models are found to have outperformed the baseline models. RF Parameters' bounds and best optimized parameters using the PSO optimizer are shown in the following table (Table I):

TABLE I. RF Parameters' Bounds and Best Optimized Parameters

RF Parameter	Lower Bound (LB)	Upper Bound (UB)	Final Optimized Value (PSO-RF)
n_estimators	50	500	126
max_depth	5	30	27
min_samples_split	2	20	2
min_samples_leaf	1	10	1
max_features	0.3	1.0	0.951048340383684

RF Optimization resolves black box Interpretation gap: In our model, RF feature selection is used, which prevents dropping domain-critical climate features. RF Optimization resolves the static hyperparameter gap: Our framework addresses this gap by optimizing the RF model using PSO, GWO, WOA, and HHO optimizers, which optimize the RF model based on predefined Lower and Upper Bounds of the RF model's parameters.

PSO-optimized RF performed the best with R^2 Value= 98.32. Fig. 10 represents the comparison of models based on R^2value.

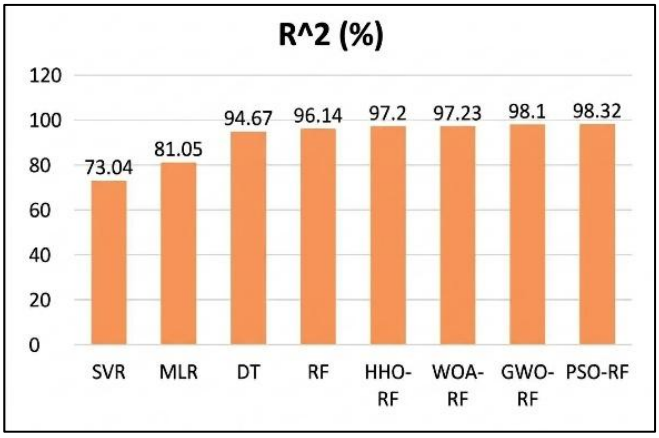


Fig. 10. Comparison of models based on R^2 value.

All four of the optimized models are found to have outperformed the baseline models. And PSO Optimized RF performed the best with CV_Score =97.22. Fig. 11 represents the comparison of models based on the CV_Score value.

All four of the optimized models are found to have outperformed the baseline models. And PSO Optimized RF performed the best with MAE =9.2766. Fig. 12 represents the comparison of models based on MAE value.

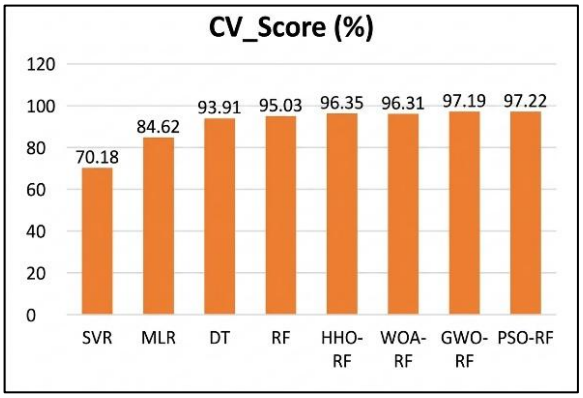


Fig. 11. Comparison of models based on CV_Score value.

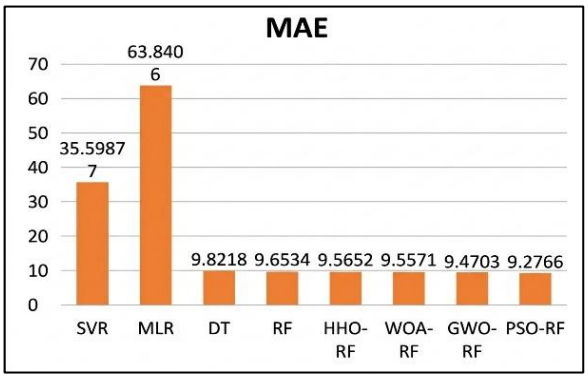


Fig. 12. Comparison of models based on MAE value.

It is observed that RMSE value is quite large. It is due to the dataset, as it contains many samples with 0 or extremely low crop yield. We cannot delete these samples, as it shows that during that period crop yield was very low or the crop was completely wasted, and we want to maintain the originality of the dataset. These yield may be low due to lack of water or weather condition or may be due to pest in the crop or due to weed or due to disaster or may be some other reason. Our algorithms focus on the other samples and predicted low or 0 crop yield values with high error. These samples are few; that is why RMSE and MSE are highly impacted, but have very little impact on R^2. All four of the optimized models are found to have outperformed the baseline models. And PSO-optimized RF performed the best with RMSE value = 116.1088. Fig. 13 represents the comparison of models based on RMSE value.

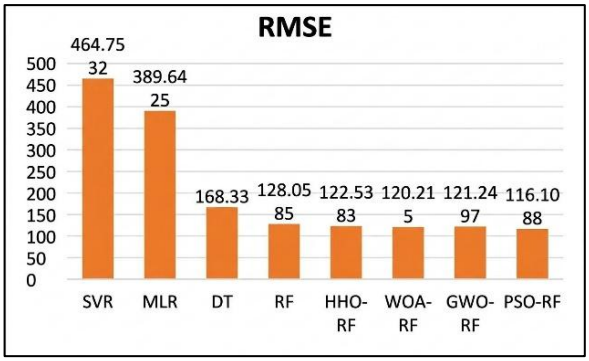


Fig. 13. Comparison of models based on RMSE value.

A. Comparison of Optimized RF Models with Base Models

Tabular comparison of baseline models and Optimized RF models is shown in Table II.

TABLE II. COMPARISON USING PERFORMANCE METRICS

Model	R^2 %	CV_Score (%)	MAE	RMSE
SVR	73.04	70.18	35.5987	464.7532
MLR	81.05	84.62	63.8406	389.6425
DT Regression	94.67	93.91	9.8218	168.3300
RF Regression	96.14	95.03	9.6534	128.0585
HHO-RF	97.20	96.35	9.5652	122.5383
WOA-RF	97.23	96.31	9.5571	120.2150
GWO-RF	98.10	97.19	9.4703	121.2497
PSO-RF	98.32	97.22	9.2766	116.1088

VI. CONCLUSION AND FUTURE SCOPE

In this research work, a data analytic framework for crop yield prediction is proposed with a comparison between four baseline models and four optimized RF models. The framework is based on the impact of climate change. The proposed framework mainly comprises of 5 phases: Data input; Data preprocessing; Feature selection based on the RF approach; Train and Test split; After the split all four baseline models are trained and perform crop yield prediction with the computation of four performance metrics: R^2, MAE, RMSE; 5 Fold CV score is computed; next, out of the four baseline models RF model is chosen as it performed best. RF model is further improved using four nature-inspired models: PSO, GWO, HHO, and WOA. It is observed that all four optimized RF models performed better than the four baseline models. And out of four optimized RF models, PSO optimized performed the best. Therefore, the RF-PSO hybrid model is proposed. The proposed framework's R^2 (%) value=98.32, MAE value= 9.2766, CV score (%) =97.22, RMSE=116.1088. High RMSE is due to samples with 0 crop yield or very low crop yield. Thus, the proposed framework is suitable for crop yield detection using RF optimization with the PSO optimization technique. In the future, the latest modern metaheuristics can be applied for further improvement in the model.

REFERENCES

[1] Calvin, K.; Fisher-Vanden, K. Quantifying the indirect impacts of climate on agriculture: An inter-method comparison. Environ. Res. Lett. 2017, 12, 115004.

[2] Prospects by Major Sector (2020, April 10). http://www.fao.org/3/Y3557E/y3557e08.htm

[3] Wheeler, T.; Von Braun, J. Climate change impacts on global food security. Science 2013, 341, 508–513. Lesk, C.; Rowhani, P.; Ramankutty, N. Influence of extreme weather disasters on global crop production. Nature 2016, 529, 84–87.

[4] Abbe, C. The physical basis of long-range weather. Mon. Weather Rev. 1901, 29, 551–561.

[5] Modi, A. · Sharma, P. Saraswat, D. Mehta R. Review of crop yield estimation using machine learning and deep learning techniquesScalable Computing. 2022; 23(2):59-79

[6] Van Klompenburg, T. Kassahun, A. · Catal, C.Crop yield prediction using machine learning: a systematic literature reviewComput. Electron. Agric. 2020; 177(July), 105709

[7] Samuel, A.L. Some Studies in Machine Learning Using the Game of Checkers. IBM J. Res. Dev. 1959, 44, 206–226. [CrossRef]

[8] Azuaje, F. Review of " data mining : practical machine learning tools and techniques " by witten and frank. January 2006 https://doi.org/10.1186/1475-925X-5-51 Date: 2014

[9] Machine Learning in Agriculture: A Review Konstantinos G. Liakos 1 , PatriziaBusato 2, DimitriosMoshou 1,3, Simon Pearson 4 ID and DionysisBochtis 1

[10] Singh, K. K., Baxla, A. K., Singh, P., & Singh, P. K. (2019). Weather based information on risk management in agriculture. In Climate change and agriculture in India: Impact and Adaptation (pp. 207-216). Springer, Cham.

[11] Ramesh, V., & Kumaresan, P. (2026). Hybrid bio-inspired optimization and Q-learning framework for deep learning crop yield prediction. Array, 30, 100982.

[12] Mohanty N. K., Mallick P.K.and Tripathy H. K., Climate-Sensitive Modeling of Rice Stock in Odisha Using Hybrid Machine Learning and Explainable AI, in IEEE Access, vol. 14, pp. 28785-28805, 2026, doi: 10.1109/ACCESS.2026.3665921.

[13] Chaurasia A., Kaur A.,A Hybrid Machine Learning Approach for Accurate Crop Yield Forecasting, Multimodal Artificial Intelligence in Precision Agriculture, Edition1st Edition, 2026, CRC Press

[14] Manjunath, M. C., & Palayyan, B. P. (2023). An efficient crop yield prediction framework using hybrid machine learning model. Revue d'Intelligence Artificielle, 37(4), 1057.

[15] Dehghanisanij, H., Emami, H., Emami, S. et al. A hybrid machine learning approach for estimating the water-use efficiency and yield in agriculture. Sci Rep 12, 6728 (2022). https://doi.org/10.1038/s41598-022-10844-2

[16] Juan Cao, Zhao Zhang, Yuchuan Luo, Liangliang Zhang, Jing Zhang, Ziyue Li, Fulu Tao, Wheat yield predictions at a county and field scale with deep learning, machine learning, and google earth engine, European Journal of Agronomy, Volume 123, 2021, 126204, ISSN 1161-0301, https://doi.org/10.1016/j.eja.2020.126204.

[17] Saeed Khaki, Lizhi Wang, and Sotirios V Archontoulis. A cnn-rnn framework for crop yield prediction. Frontiers in Plant Science, 10:1750, 2020. Zhang L., Zhang J., Li Z., Tao F., Wheat yield predictions at a county and field scale with deep learning, machine learning, and google earth engine, European Journal of Agronomy, Volume 123, 2021, 126204, ISSN 1161-0301, https://doi.org/10.1016/j.eja.2020.126204.

[18] Jichong Han, Zhao Zhang, Juan Cao, Yuchuan Luo, Liangliang Zhang, Ziyue Li, and Jing Zhang. Prediction of winter wheat yield based on multi-source data and machine learning in china. Remote Sensing, 12(2):236, 2020.

[19] Jie Sun, Liping Di, Ziheng Sun, Yonglin Shen, and Zulong Lai. Countylevel soybean yield prediction using deep cnn-lstm model. Sensors, 19(20):4363, 2019

[20] PS Maya Gopal and R Bhargavi. A novel approach for efficient crop yield prediction. Computers and Electronics in Agriculture, 165:104968, 2019.

[21] Yousif Y Hilal, Wan Ishak, Azmi Yahya, and Zulfa H Asha'ari. Development of genetic algorithm for optimization of yield models in oil palm production. Chilean journal of agricultural research, 78(2):228– 237, 2018

[22] Alberto González Sánchez, Juan Frausto Solís, Waldo Ojeda Bustamante, et al. Predictive ability of machine learning methods for massive crop yield prediction. Spanish Journal Of Agricultural Research, 12(2):313–328, 2014.

[23] Ruan, G., Li, X., Yuan, F., Cammarano, D., Ata-UI-Karim, S. T., Liu, X., Tian, Y., Zhu, Y., Cao, W., Cao, Q. (2022). Improving wheat yield prediction integrating proximal sensing and weather data with machine learning. Computers and Electronics in Agriculture, 195, 106852.

[24] Yue, Y., Li, J. H., Fan, L. F., Zhang, L. L., Zhao, P. F., Zhou, Q., ... & Dong, X. H. (2020). Prediction of maize growth stages based on deep learning. Computers and electronics in agriculture, 172, 105351.

[25] Quinlan, J. R. (1986). "Induction of decision trees" (PDF). Machine Learning. 1: 81–106. doi:10.1007/BF00116251. S2CID 189902138.

[26] von Winterfeldt, Detlof; Edwards, Ward (1986). "Decision trees". Decision Analysis and Behavioral Research. Cambridge University Press. pp. 63–89. ISBN 0-521-27304-8.

[27] Yan, Xin (2009), Linear Regression Analysis: Theory and Computing, World Scientific, pp. 1–2, ISBN 978-981-283-411-9, archived from the original on 2024-10-04, retrieved 2015-02-07

[28] Drucker, Harris; Burges, Christ. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); "Support Vector Regression Machines", in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

[29] Prasad, N. R., Patel, N. R., &Danodia, A. (2021). Crop yield prediction in cotton for regional level using random forest approach. Spatial Information Research, 29(2), 195-206.

[30] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. Vol. IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[31] Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73. doi:10.1109/ICEC.1998.699146.

[32] Mirjalili, Seyedali; Mirjalili, Seyed Mohammad; Lewis, Andrew (2014). "Grey Wolf Optimizer". Advances in Engineering Software. 69. ElsevierBV: 46– 61. doi:10.1016/ j.advengsoft.2013.12.007. hdl:10072/66188. ISSN 0965-9978. Retrieved 2025-04-02.

[33] Qiu, Yihui; Yang, Xiaoxiao; Chen, Shuixuan (2024-06-20). "An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems" (PDF). Scientific Reports. 14 (1). Springer Science and Business Media LLC: 14190. Bibcode:2024NatSR..1414190Q. doi:10.1038/s41598-024-64526-2. ISSN 2045-2322.

[34] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, Huiling Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems, Volume 97, 2019, Pages 849-872, ISSN 0167-739X, https://doi.org/10.1016/j.future.2019.02.028

[35] Seyedali Mirjalili, Andrew Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, Volume 95, 2016, Pages 51-67, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2016.01.008.

[36] https://www.kaggle.com/datasets/akshatgupta7/crop-yield-in-indian-states-dataset

Student-Disjoint Cross-Module Benchmarking for Calibrated Early Warning in Online Learning

Mika Lim
University Innovation Center, Seoul National University, Seoul, Republic of Korea

***Abstract*—Early-warning models in online learning are commonly evaluated with random data partitions, even though deployment often requires transfer to a course that supplied no training data. This study establishes a leakage-resistant benchmark for cross-module risk prediction using the Open University Learning Analytics Dataset. The analysis covered 32,593 enrollments from 28,785 learners, seven modules, 22 module presentations, and 10,655,280 daily virtual learning environment interaction records. Predictions were produced after 7, 14, and 28 days for learners still active at each cutoff. Logistic regression, random forest, and histogram gradient boosting were evaluated by leaving one entire module out and removing every learner who also appeared in the held-out module from training. Probability calibration was learned only from student-grouped out-of-fold predictions, using the same nested procedure in both validation designs. Random student-disjoint cross-validation yielded an area under the receiver operating characteristic curve (AUROC) of 0.716, whereas held-out-module evaluation yielded an AUROC of 0.661, revealing substantial contextual optimism. Under the fixed hyperparameter configurations used in this benchmark, logistic regression achieved higher cross-module performance than both nonlinear models. A virtual-learning-environment-only specification preserved discrimination while improving precision-recall performance and Brier score relative to adding assessment-submission features. Calibration nevertheless varied markedly across modules, indicating that global post-hoc calibration did not eliminate course-level distribution shift. The benchmark shows that student independence, course separation, feature availability, and probability quality must be evaluated jointly before an early-warning model is considered transferable.**

***Keywords*—Learning analytics; educational data mining; early warning; student risk prediction; cross-course generalization; probability calibration; data leakage; online learning**

I. Introduction

Learning management systems record how learners access resources, participate in activities, and submit assessments. These traces support early-warning models that prioritize timely academic support. Research using the Open University Learning Analytics Dataset (OULAD) and other large online-learning corpora has shown that behavioral variables can predict failure or withdrawal before a course ends [1]–[4]. Transfer to a new course, however, is a different problem from predicting held-out learners sampled from the same mixture of courses. Course organization, assessment timing, resource structure, and outcome prevalence can change the relationship between a feature and the target. A model may therefore appear accurate in a random split yet fail when used in an unseen instructional context.

The distinction matters technically and educationally. Repeated learners can enroll in more than one module, allowing student-level regularities to cross a nominal train-test boundary. Random partitions also mix course contexts, so the test distribution resembles the training distribution more closely than a future deployment course. Leakage and distribution mismatch can inflate claims in machine-learning-based science [10]. Recent learning-analytics research has consequently emphasized transfer, external validity, and model stability across courses and cohorts [5]–[9]. Yet many studies still optimize discrimination alone. Early-warning systems use predicted probabilities to allocate scarce support; poor calibration can over- or under-estimate the number of learners requiring intervention even when rank discrimination is acceptable [11], [12].

A second concern is feature legitimacy. Assessment scores can be highly predictive, but OULAD reports submission dates rather than the time at which grades became available. Treating every recorded score as known at the prediction cutoff can therefore introduce timing ambiguity. Demographic variables may improve average performance while creating governance and fairness concerns. Course-specific assessment counts and due-date patterns may also transfer poorly because they encode instructional design rather than general engagement. A deployable benchmark should distinguish registration data, virtual learning environment (VLE) behavior, submission behavior, recorded scores, and protected or sensitive attributes instead of aggregating them into an undifferentiated feature set.

The computational contribution of this study is an evaluation protocol rather than a new classifier: a reproducible student-disjoint, leave-one-module-out benchmark for calibrated early-warning prediction on all seven OULAD modules. It makes four contributions. First, it separates both modules and learner identities between outer training and test sets. Second, it evaluates three prediction horizons while defining a dynamic risk set of learners who remain enrolled at each cutoff. Third, it compares feature families according to operational availability and transferability. Fourth, it evaluates probability quality, module heterogeneity, paired model differences, and model-agnostic feature importance rather than reporting a single accuracy value. The technical novelty lies in coupling these safeguards within a single transfer benchmark rather than evaluating them independently: learner identities and modules are separated at the outer boundary, the risk set is redefined at each prediction cutoff, calibration is learned entirely within training data, and feature families are compared under the same deployment-oriented protocol.

RQ1. How do validation design, prediction horizon, and model class affect cross-module discrimination and probability quality?

RQ2. Which feature families provide the most transferable balance of discrimination and calibration for an unseen module?

RQ3. How stable are calibrated risks across held-out modules, outcomes, and time, and which observed behaviors drive the predictions?

II. Related Work

A. Early Risk Prediction from Learning Traces

OULAD links demographics, registration, assessments, course structure, and daily VLE interactions for seven anonymized modules [1]. Deep and ensemble models have achieved strong within-dataset prediction, and flexible feature-selection methods can reduce manual engineering [2], [3]. Higher temporal granularity often improves early prediction [4]. These findings establish the predictive value of activity traces, but a model trained and tested within a pooled course mixture answers a narrower question than transfer to a previously unseen module. High apparent accuracy may also depend on final-course information, assessment schedules, or learners represented in both partitions.

B. Cross-Course Generalization

Transfer-learning research has trained models across multiple massive open online courses and used course metadata or prior iterations to warm-start prediction [5]. Other work shows that generalizability depends on course and cohort context and that behavioral models can lose stability after transfer [6], [13]. Recent studies have evaluated digital-trace models across computer-science courses [7] and multiple technical-education courses [8], while modelability has been proposed as a design principle for scalable predictive learning analytics [9]. The present study complements this literature by combining course-disjoint testing with complete learner-identity separation, multiple early cutoffs, feature-availability ablations, and calibration analysis on a widely reused public benchmark.

C. Calibration, Errors, and Transparency

The area under the receiver operating characteristic curve (AUROC) measures ranking, not whether a predicted risk of 0.60 corresponds to an observed event rate near 0.60. Modern classifiers can be miscalibrated even when discrimination is strong [14]. Platt scaling provides a simple post-hoc mapping from model scores to probabilities [15], while Brier score [17] and expected calibration error [14] summarize probability accuracy. Because non-success is common but not balanced uniformly across modules, area under the precision-recall curve is also necessary [16]. Educational consequences further require examination of errors and explanation: confident mistakes can erode adoption, and important contextual causes of non-success may remain absent from platform data [12]. Transparent comparisons between simple and complex models are therefore preferable to assuming that higher-capacity algorithms will generalize better [21].

III. Methods

A. Data Source and Integrity Checks

The study used OULAD, released under a Creative Commons Attribution 4.0 license [1]. The seven relational tables were joined by module, presentation, learner, assessment, and VLE-site identifiers. Before modeling, record counts and module-presentation coverage were checked against the data descriptor. The verified data contained 32,593 enrollment records, 28,785 unique learners, seven modules, 22 presentations, 206 assessments, 6,364 VLE sites, and 10,655,280 student-VLE daily interaction records. No duplicate enrollment keys or missing outcomes were present.

The binary target was non-success, defined as a final result of Fail or Withdrawn; Pass and Distinction formed the comparison class. At each prediction day, learners whose recorded unregistration date preceded the cutoff were excluded because they were no longer candidates for prospective support. This dynamic risk-set definition produced 29,196 active enrollments at day 7, 28,128 at day 14, and 27,553 at day 28. The positive prevalence declined from 0.473 to 0.442 as already-withdrawn learners left the active set. A fixed-cohort sensitivity analysis restricted all three horizons to the 27,553 enrollments still active at day 28. The active cohorts and corresponding outcome prevalence are summarized in Table I.

TABLE I. Active Cohorts and Outcome Prevalence

Day	Active n	Non-success n	Prevalence
7	29,196	13,811	0.473
14	28,128	12,743	0.453
28	27,553	12,168	0.442

B. Feature Construction

All time-varying features used records dated from course day 0 through the day preceding the cutoff, so no same-day or future event entered a prediction. Registration variables comprised registration timing, previous attempts, and studied credits. VLE features summarized total, recent, and first-week clicks; active days and their ratio to elapsed days; daily mean, standard deviation, and maximum; number of unique resources; recent-to-first-week ratio; and clicks within 14 activity categories. Nonnegative skewed counts were transformed by $\log(1+x)$. Submission-behavior features counted submitted and late assessments, indicated whether any assessment had been submitted, and summarized timing relative to due dates.

Five feature specifications were evaluated. Registration used three static variables. VLE added 23 module-agnostic activity features. Operational added four assessment-submission variables but no scores. Score-augmented added mean, minimum, and maximum recorded score as a timing-sensitivity analysis. Demographic-augmented added gender, region, prior education, deprivation band, age band, and disability to the operational set. The primary transfer interpretation emphasized the VLE and operational specifications; scores and demographics were not treated as required deployment inputs.

Preprocessing was fit only on each training set. Numeric missing values used the training median, counts used structural zeros where no event occurred, and demographic categories used the training mode plus one-hot encoding.

C. Models and Calibration

Three classifiers represented increasing functional complexity: L2-regularized logistic regression, a random forest with 160 trees and a minimum leaf size of 10 [18], and histogram gradient boosting with 180 iterations, learning rate 0.08, 31 leaf nodes, and L2 regularization. Hyperparameters were fixed across modules and cutoffs to avoid adapting the benchmark to a held-out target. Accordingly, comparisons among model classes refer to these prespecified configurations and should not be interpreted as an exhaustive test of each algorithm after hyperparameter optimization. Model-class comparisons used the operational specification at all three cutoffs. Day-28 feature-family analyses used logistic regression, the strongest cross-module model in that comparison, to isolate feature effects without multiplying the model grid. Logistic numeric inputs were standardized within the training pipeline. Implementations used scikit-learn [19] with a fixed random seed.

For each outer fold, 4-fold stratified group cross-validation generated out-of-fold predictions within the training modules, with learner identifier as the group. A Platt logistic calibrator was fit to those out-of-fold scores and then applied to the untouched held-out module [15]. The F1 threshold was selected from calibrated training predictions only. Uncalibrated and calibrated probabilities were retained; threshold-dependent results were secondary because intervention capacity was not specified.

D. Validation Design and Metrics

The principal evaluation used seven leave-one-module-out folds. In each fold, all enrollments from one module were held out. Every enrollment of a learner appearing in that test module was then removed from the remaining training modules. Depending on the fold, this identity filter removed 2 to 1,772 otherwise eligible training rows. This design estimates transfer to a new module for learners unseen during model fitting. A student-disjoint stratified 5-fold split that ignored module boundaries served as the conventional comparator. The comparator used the same 4-fold, student-grouped inner procedure for Platt calibration, ensuring that the contrast changed the module boundary rather than the calibration protocol.

Discrimination was assessed with AUROC and area under the precision-recall curve (AUPRC). Probability quality was assessed with Brier score, log loss, and expected calibration error computed with ten equal-frequency bins. Accuracy, balanced accuracy, precision, recall, and F1 were reported for training-derived thresholds. Student-cluster bootstrapping preserved repeated enrollments when calculating 95 percent intervals and paired model differences. Permutation importance was computed within each held-out module as the mean decrease in AUROC over three independent permutations of each feature. No normalization was applied to the importance values. Module-specific importance estimates were then aggregated using held-out-module sample sizes as weights. Negative permutation importance values were retained without truncation or transformation; Fig. 5 displays only the nine highest-ranked features.

IV. Results

A. Model and Horizon Comparison

Under the prespecified hyperparameter configurations, logistic regression achieved the highest pooled cross-module AUROC at every horizon. With operational features, calibrated AUROC increased from 0.654 at day 7 to 0.667 at day 14 and then decreased slightly to 0.661 at day 28 (Fig. 1 and Table II). The corresponding day-28 AUPRC, Brier score, and expected calibration error were 0.594, 0.237, and 0.071; AUPRC exceeded the active-cohort prevalence baseline of 0.442. The student-cluster bootstrap interval for day-28 AUROC was 0.655 to 0.667. Under these fixed configurations, random forest and histogram gradient boosting showed lower transfer performance than logistic regression. At day 28, logistic regression exceeded random forest by 0.0136 AUROC (95 percent interval 0.0104 to 0.0170) and histogram gradient boosting by 0.0082 (0.0050 to 0.0115). Its Brier score was lower by 0.0063 and 0.0060, respectively, with both paired intervals excluding zero.

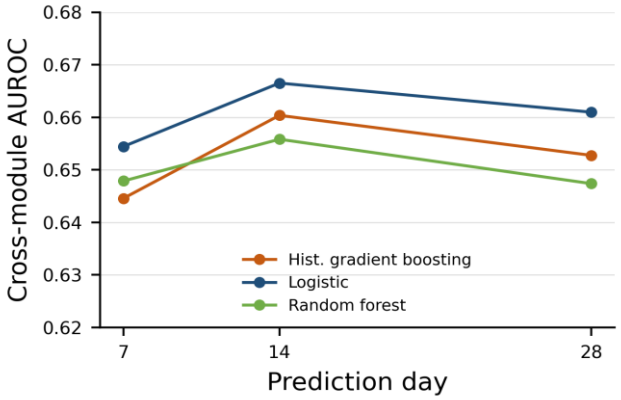


Fig. 1. Student-disjoint held-out-module AUROC by prediction day and model using operational features.

TABLE II. Cross-Module Performance with Operational Features

Day	Model	AUROC	AUPRC	Brier
7	Logistic	0.654	0.611	0.232
7	RF	0.648	0.598	0.234
7	HGB	0.645	0.600	0.234
14	Logistic	0.667	0.615	0.229
14	RF	0.656	0.596	0.231
14	HGB	0.660	0.604	0.230
28	Logistic	0.661	0.594	0.237
28	RF	0.647	0.575	0.243
28	HGB	0.653	0.574	0.243

RF: random forest; HGB: histogram gradient boosting. Values use Platt-calibrated probabilities.

B. Validation-Design Gap

With the same nested calibration protocol, the conventional student-disjoint 5-fold design, which mixed modules across

folds, produced AUROC 0.716, AUPRC 0.673, Brier score 0.211, and expected calibration error 0.017 at day 28. The stricter student-disjoint held-out-module design produced 0.661, 0.594, 0.237, and 0.071, respectively (Fig. 2). Thus, merely preventing learner overlap did not simulate deployment to an unseen module. Module separation exposed an AUROC optimism gap of 0.055 and a more than fourfold increase in calibration error. This result answers the validation component of RQ1: context mixing remained a major source of optimistic evaluation after learner identities were separated.

The magnitude of this gap should not be treated as directly equivalent to prior transfer results because the prediction targets, datasets, and validation schemes differ across studies. Kleter et al. [6] evaluated both numerical and binary predictive models across courses and cohorts and reported performance degradation after generalization, with the extent of degradation depending on initial model performance. Schwerter et al. [7] reported in-sample AUROC ranges of 0.73–0.81 for Elastic Net, 0.94–1.00 for random forest, and 0.77–0.99 for XGBoost, while cross-course and cross-institution performance declined and varied substantially across source-target pairings. By contrast, the present 0.716-to-0.661 comparison holds the dataset, outcome definition, feature specification, learner-disjoint rule, and nested calibration procedure constant, so the 0.055 difference more directly isolates the effect of enforcing module separation.

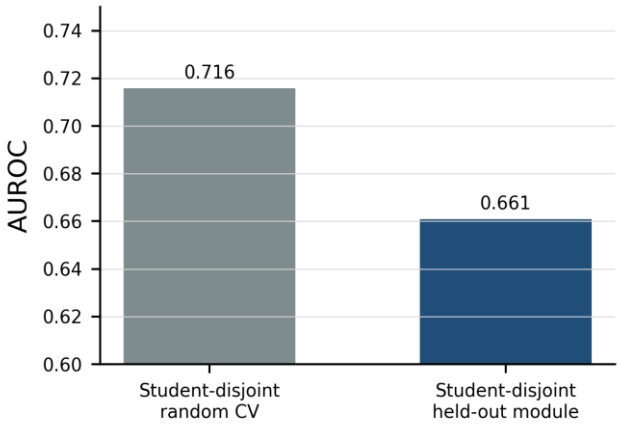


Fig. 2. Validation design changes apparent day-28 logistic performance despite student-disjoint partitions.

C. Feature Transferability

Feature-family results showed that additional information was not uniformly transferable (Table III and Fig. 3). Registration data alone provided weak discrimination. VLE features increased AUROC to 0.660 and AUPRC to 0.604 while achieving the lowest Brier score among the behavior-based models, 0.230. Adding submission behavior changed AUROC by less than 0.001 but reduced AUPRC by 0.010 and increased Brier score by 0.0068. In paired student-cluster bootstrap comparisons, the VLE-minus-operational AUROC difference was -0.0009 (95 percent interval -0.0055 to 0.0036), whereas its AUPRC advantage was 0.0102 (0.0057 to 0.0145) and its Brier difference was -0.0068 (-0.0085 to -0.0050). The latter two improvements were robust.

Recorded scores raised AUROC to 0.672 but reduced AUPRC and worsened Brier score. Demographics yielded the highest AUROC, 0.684, and AUPRC, 0.619, but used six sensitive or governance-relevant variables. These sensitivity results do not establish that such variables should be deployed. Rather, they demonstrate that feature selection changes the balance between ranking, probability quality, timing validity, and responsible use. The VLE-only specification provided the strongest transfer-oriented compromise for RQ2.

TABLE III. DAY-28 LOGISTIC FEATURE-SET ABLATION

Feature set	p	AUROC	AUPRC	Brier	ECE
Registration	3	0.551	0.507	0.243	0.015
VLE	26	0.660	0.604	0.230	0.044
Operational	30	0.661	0.594	0.237	0.071
+ Scores	33	0.672	0.574	0.238	0.084
+ Demographics	36*	0.684	0.619	0.230	0.067

p counts six demographic fields before one-hot expansion. ECE: expected calibration error.

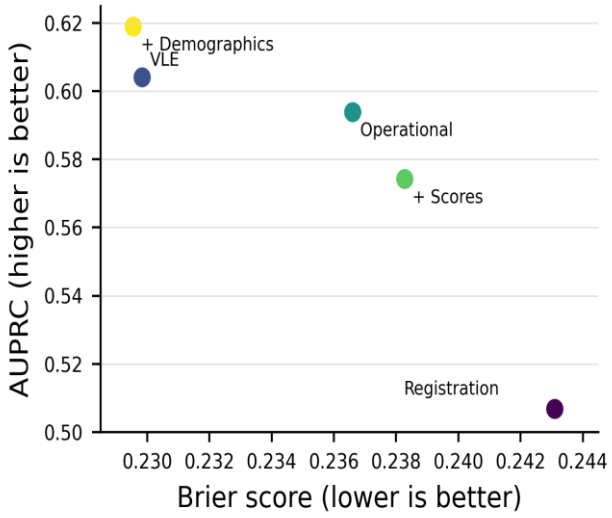


Fig. 3. Trade-off between precision-recall performance and probability accuracy across feature families.

D. Module Heterogeneity and Calibration

Held-out-module results varied substantially. For the day-28 operational logistic model, AUROC ranged from 0.635 for GGG to 0.736 for EEE. Expected calibration error ranged from 0.027 for CCC to 0.357 for GGG. The VLE-only reliability curves showed the same broader pattern: some modules lay above the identity line and others below it (Fig. 4). Global Platt scaling slightly improved pooled expected calibration error for logistic regression, from 0.073 to 0.071, but it did not correct target-specific base-rate and feature-distribution shifts. For random forest and histogram gradient boosting at day 28, Platt scaling worsened pooled Brier score and expected calibration error. Calibration therefore must be checked on each destination context rather than assumed to transfer with discrimination.

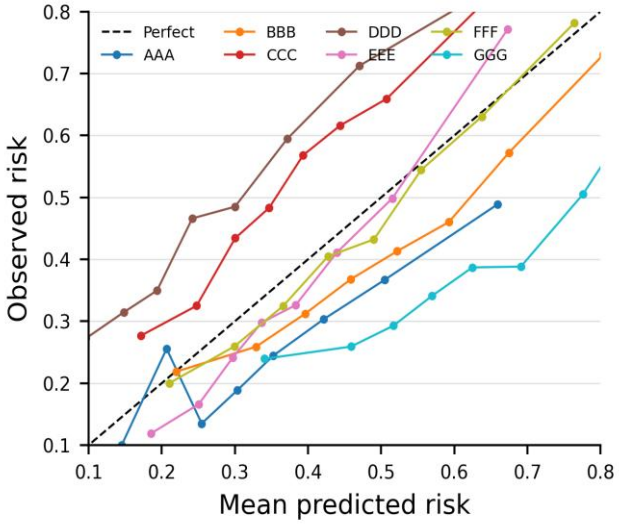


Fig. 4. Reliability of the day-28 VLE-only logistic model in each held-out module.

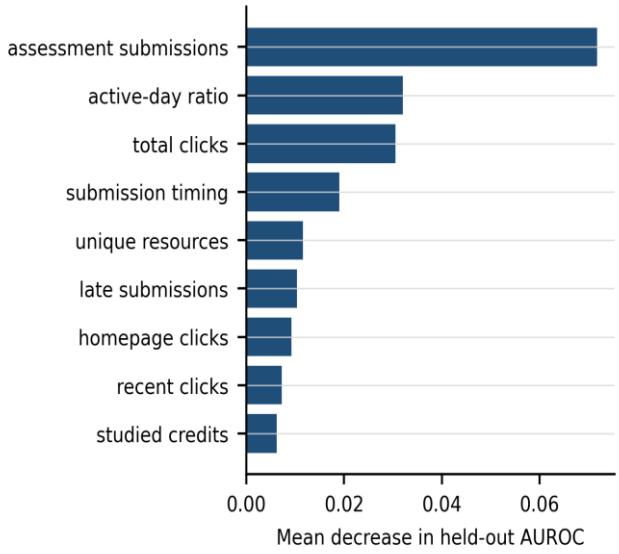


Fig. 5. Weighted permutation importance across day-28 held-out modules for operational logistic regression.

E. Time, Risk-Set Composition, and Errors

The fixed day-28 cohort analysis separated increasing observation length from attrition-induced changes in the active population. For operational logistic regression on the same 27,553 enrollments, AUROC was 0.644 at day 7, 0.665 at day 14, and 0.661 at day 28. The day-28 decline therefore was not explained solely by removal of already-withdrawn learners. For the VLE-only model, AUROC changed from 0.643 to 0.658 to 0.660, while AUPRC increased from 0.569 to 0.594 to 0.604. Submission behavior helped at day 14 but harmed day-28 probability quality, consistent with growing cross-module differences in assessment schedules. More elapsed time did not guarantee better transferable operational features.

At the training-derived decision thresholds, the operational logistic model correctly flagged 76.2 percent of eventual failures and 72.6 percent of withdrawals, but it also flagged 53.8 percent of eventual passes and 47.4 percent of distinctions. These errors caution against treating predicted risk as a deterministic judgment. The output is more appropriate for low-stakes outreach or instructor review than for exclusionary decisions.

F. Feature Importance

Permutation importance for the day-28 operational logistic model identified the number of assessment submissions as the strongest predictor, followed by active-day ratio, total clicks, submission timing, unique resources, and late submissions (Fig. 5). This result should be interpreted as within-held-out-module sensitivity rather than as evidence that feature importance is invariant across modules. The present analysis does not decompose within-module importance from cross-module stability; therefore, the weaker transfer performance observed after adding submission features is compatible with, but does not by itself establish, context-specific importance. A direct within-versus-cross-module importance decomposition would be required to test that mechanism.

V. Discussion

A. Principal Findings

The benchmark provides three answers. For RQ1, both learner identity and course context affected evaluation. Student-disjoint random cross-validation remained optimistic because modules were mixed across folds. The held-out-module design reduced AUROC by 0.055 and exposed materially worse calibration. Under the prespecified hyperparameter settings, logistic regression showed more reliable transfer than the two nonlinear models. This pattern is consistent with prior evidence that strong within-context performance does not guarantee cross-context generalization [6], [7], [10], but the present comparison cannot separate model-class effects from possible undertuning of the nonlinear models. The result does not imply that logistic regression is universally superior; it shows that model ranking depends on the deployment distribution represented by the validation design.

For RQ2, the VLE-only model offered a more defensible transfer point than the broader operational model. Submission variables were individually important, yet adding them reduced cross-module AUPRC and probability accuracy at day 28. Course-specific deadlines, assessment counts, and marking designs can change what a submission means. This is a negative-result contribution: predictive information can be locally useful while degrading transportability. Recorded scores and demographics altered performance again, but their timing and governance costs make them sensitivity conditions rather than default inputs. A feature's availability, stability, and policy acceptability should therefore be evaluated alongside its importance.

For RQ3, calibration and error rates varied sharply across modules. Platt scaling improved pooled logistic metrics only modestly and could not correct module-specific distribution shifts. The GGG fold was especially miscalibrated even though learner overlap was minimal, indicating that the problem was

contextual rather than identity leakage. Furthermore, the fixed-cohort analysis showed that additional weeks did not monotonically improve every specification. The predictive system changes as both the evidence window and the set of still-active learners evolve. Early-warning evaluation should therefore be framed as repeated risk-set prediction rather than a single static classification task.

B. Implications for Learning Analytics Systems

A practical implementation should treat transfer as a gated software lifecycle rather than a one-time model export. A source-trained model can produce provisional risk rankings for a new module, but its probability scale should be monitored against emerging local outcomes. Until enough labeled target data are available, institutions should communicate risk bands or prioritized outreach rather than precise probabilities. When a previous presentation exists, recalibration and threshold selection should use that presentation without contaminating the target evaluation. The benchmark also supports a parsimonious default: engagement variables that have comparable definitions across courses are preferable to assessment and demographic variables whose semantics or acceptability change by context. This sequence—identity-safe validation, destination monitoring, local recalibration, and threshold governance—translates the benchmark into testable requirements for learning-analytics software.

The educational interpretation must remain limited. Clicks and submissions are traces of platform interaction, not direct measures of knowledge, motivation, or ability. False positives may reflect strategic low-click study, prior knowledge, offline work, or inaccessible contextual information; false negatives may reflect sudden work, family, health, or technical events [12]. Explanations should therefore support conversation and resource allocation, not label learners as inherently likely to fail. This human-in-the-loop framing is consistent with calls for transparent, calibrated, and ethically governed predictive analytics [11], [20]–[22].

C. Methodological Contribution

The principal contribution is not a new classifier but a stricter, reusable benchmark specification. It defines the estimand as performance on enrollments from a module absent during training; removes all cross-boundary learner identities; fits preprocessing, calibration, and thresholds inside training data; applies the same nested calibration protocol to the conventional comparator; evaluates active learners at explicit temporal cutoffs; separates feature families by operational legitimacy; and reports discrimination, probability quality, heterogeneity, uncertainty, and explanation. This combination is necessary because a standard dataset does not by itself standardize a valid evaluation [10]. The accompanying scripts produce the cohort, features, folds, predictions, tables, and figures from the verified OULAD tables. The contribution therefore lies in the joint specification of these components: none is claimed as novel in isolation, but their integration makes the deployment estimand, leakage controls, calibration boundary, temporal risk set, and feature-availability constraints explicit within one reproducible benchmark.

D. Limitations

OULAD comes from one distance-learning institution and historical course offerings. Module labels and subject content are anonymized, limiting explanations of why particular modules shifted. VLE records are daily aggregates and omit offline study, motivation, communication outside the platform, and unexpected life events. The final result is a useful operational label but does not measure learning quality or the causal effect of intervention. The evaluated algorithms and fixed hyperparameters do not represent an exhaustive search for maximum predictive performance. Consequently, the lower transfer performance of random forest and histogram gradient boosting cannot be separated from the possibility that those prespecified configurations were undertuned for this setting. The study did not tune models for each target module, use target labels for recalibration, or evaluate interventions. These restrictions make the benchmark conservative but do not establish performance at another institution. Moreover, seven held-out modules are sufficient for a benchmark but not for estimating uncertainty over the wider population of courses; the learner-cluster intervals quantify sampling variation among observed learners, not between-course uncertainty. Expected calibration error is also bin-dependent. Finally, the score-augmented analysis cannot determine exactly when a grade became visible; it should not be interpreted as a prospective score-based deployment result. The feature-importance analysis is also descriptive: it summarizes predictive sensitivity within held-out modules but does not estimate a formal within-module versus cross-module importance decomposition.

VI. Conclusion

Within the OULAD setting examined here, evaluation on an unseen module produced materially different conclusions from student-disjoint cross-validation that mixed module contexts. Student-disjoint random cross-validation substantially overstated performance relative to a student-disjoint held-out-module benchmark even when both designs used the same nested calibration procedure. Under the prespecified hyperparameter configurations, logistic regression transferred better than the evaluated nonlinear models, and VLE-only engagement features achieved nearly identical discrimination with better precision-recall and probability accuracy than a broader assessment-aware specification. Calibration remained module-dependent, so a global correction was insufficient. The resulting evaluation architecture shifts attention from maximizing a pooled score to establishing whether a model is independent, transferable, calibrated, interpretable, and based on legitimate information. These conclusions are limited to seven anonymized modules from a single distance-learning institution and should not be interpreted as evidence of performance across institutions or course populations more broadly. Future work should replicate the benchmark on at least one independent institutional dataset, incorporate systematic hyperparameter tuning within the training boundary, and test whether feature-importance patterns remain stable across source and destination modules.

Data and Code Availability

OULAD is publicly available under the Creative Commons Attribution 4.0 license through the dataset DOI https://doi.org/10.24432/C5KK69 and is described in [1]. The verified analysis used 10,655,280 student-VLE records spanning all 22 module presentations. Reproducible scripts and exact aggregate outputs are supplied with this manuscript; the source data are not redistributed.

Ethics Statement

This secondary analysis used an anonymized public dataset and involved no participant contact, intervention, or attempt at re-identification. Predictions were evaluated as research outputs and were not used to make decisions about individual learners.

Declaration on Generative AI

An OpenAI generative AI tool was used for programming assistance, manuscript drafting, language refinement, and document formatting. The human author specified the research problem and retains responsibility for independently verifying the data, code, analyses, citations, interpretation, and all AI-assisted text before submission.

References

[1] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics dataset," Scientific Data, vol. 4, art. 170171, 2017, doi: 10.1038/sdata.2017.171.

[2] H. Waheed, S. U. Hassan, N. R. Aljohani, B. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," Computers in Human Behavior, vol. 104, art. 106189, 2020, doi: 10.1016/j.chb.2019.106189.

[3] A. Al-Zawqari et al., "A flexible feature selection approach for predicting students' academic performance in online courses," Computers and Education: Artificial Intelligence, vol. 3, art. 100103, 2022, doi: 10.1016/j.caeai.2022.100103.

[4] C. Ye and G. Biswas, "Early prediction of student dropout and performance in MOOCs using higher granularity temporal information," Journal of Learning Analytics, vol. 1, no. 3, pp. 169–172, 2014, doi: 10.18608/jla.2014.13.14.

[5] V. Swamy, M. Marras, and T. Käser, "Meta transfer learning for early success prediction in MOOCs," in Proc. 9th ACM Conf. Learning at Scale, 2022, pp. 121–132, doi: 10.1145/3491140.3528273.

[6] S. Kleter, U. Matzat, and R. Conijn, "Boundary conditions of generalizing predictive models for academic performance: Within cohort versus within course," IEEE Transactions on Learning Technologies, vol. 17, pp. 2183–2194, 2024, doi: 10.1109/TLT.2024.3443079.

[7] J. Schwerter et al., "Cross-course generalizability of SRL-aligned predictive models using digital learning traces," Computers & Education, vol. 253, art. 105670, 2026, doi: 10.1016/j.compedu.2026.105670.

[8] O. Ovtšarenko, "Using AI to forecast student dropout risk in technical education using a learning analytics approach," Scientific Reports, vol. 16, art. 14616, 2026, doi: 10.1038/s41598-026-44919-1.

[9] A. Xu, Y. Zhang, A. B. Blake, and J. W. Stigler, "Modelability as a strategy for improving the generalizability and scalability of predictive models," Journal of Learning Analytics, vol. 13, no. 1, pp. 89–109, 2026, doi: 10.18608/jla.2026.9099.

[10] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, art. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

[11] A. Mathrani, S. Sarvesh, and T. Umer, "Perspectives on the challenges of generalizability, transparency and ethics in predictive learning analytics," Computers and Education Open, vol. 2, art. 100060, 2021, doi: 10.1016/j.caeo.2021.100060.

[12] M. Hlosta, C. Herodotou, T. Papathoma, A. Gillespie, and P. Bergamin, "Predictive learning analytics in online education: A deeper understanding through explaining algorithmic errors," Computers and Education: Artificial Intelligence, vol. 3, art. 100108, 2022, doi: 10.1016/j.caeai.2022.100108.

[13] R. Conijn, C. C. P. Snijders, P. A. M. Kleingeld, and U. Matzat, "Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS," IEEE Transactions on Learning Technologies, vol. 10, no. 1, pp. 17–29, 2017, doi: 10.1109/TLT.2016.2616312.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. 34th Int. Conf. Machine Learning, vol. 70, 2017, pp. 1321–1330.

[15] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, A. J. Smola et al., Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.

[16] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLOS ONE, vol. 10, no. 3, art. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[17] G. W. Brier, "Verification of forecasts expressed in terms of probability," Monthly Weather Review, vol. 78, no. 1, pp. 1–3, 1950, doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

[18] L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[20] H. Khosravi et al., "Explainable artificial intelligence in education," Computers and Education: Artificial Intelligence, vol. 3, art. 100074, 2022, doi: 10.1016/j.caeai.2022.100074.

[21] K. A. Bird, B. L. Castleman, Z. Mabel, and Y. Song, "Bringing transparency to predictive analytics: A systematic comparison of predictive modeling methods in higher education," AERA Open, vol. 7, no. 1, pp. 1–19, 2021, doi: 10.1177/23328584211037630.

[22] M. Karimi-Haghighi, C. Castillo, D. Hernández-Leo, and V. Moreno Oliver, "Predicting early dropout: Calibration and algorithmic fairness considerations," in Companion Proc. 11th Int. Conf. Learning Analytics and Knowledge (LAK21), 2021, pp. 294–304.

Dynamic Ant Colony Optimization–Driven Feature Selection for GRU-Based Network Intrusion Detection

Zhaosheng Yang
Department of Information Engineering, Shijiazhuang College of Applied Technology,
Shijiazhuang City, Hebei Province, 050800, China

***Abstract*—Modern network environments generate large-scale, high-dimensional traffic data that can be effectively interpreted as complex temporal signals. Efficiently analyzing such data requires robust feature selection and accurate temporal modeling techniques. In this study, we propose a signal-driven intrusion detection framework that integrates a Dynamic Ant Colony Optimization (DyACO) algorithm with deep recurrent neural modeling. The proposed DyACO performs adaptive wrapper-based feature selection by dynamically adjusting pheromone evaporation, heuristic guidance, and reinforcement strategies to enhance exploration in high-dimensional signal spaces. Selected features are then processed using a gated recurrent unit network to capture temporal dependencies in network traffic signals. The experiments on the CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD datasets demonstrate that our proposed method not only effectively compresses the feature space but also enhances classification accuracy, precision, recall, and F-score. The results demonstrate that when applied to network traffic, structured temporal signals, along with adaptive optimization modeling, yield an effective solution for intelligent, signal-based anomaly detection.**

Keywords*—Ant colony optimization; feature selection; network intrusion detection; deep learning

I. INTRODUCTION

Nowadays, networked environments produce enormous amounts of diverse traffic data, complicating intrusion detection [1]. Owing to the increasing volume and complexity of cyberattacks, Intrusion Detection Systems (IDSs) face the difficulty of processing highly-dimensional spaces rich in redundant, noisy, and correlated attributes [2]. These factors create challenging classification tasks, high computational requirements, and rising false-positive rates. Similar challenges have also been observed in other security-sensitive domains, where machine learning-based detection approaches require effective feature selection, preprocessing strategies, and imbalance handling mechanisms to improve detection reliability [3]. Moreover, multi-class distributions and class imbalance in standard IDS benchmark sets can further degrade performance because detection results are unbalanced across attack classes. This demonstrates the limited performance of the standard architecture of IDSs under increasing data volume [4].

Feature Selection (FS) is an important technique for improving IDS reliability in high-dimensional network traffic data. Feature Selection is a process of choosing the most relevant features of a dataset [5]. FS enhances efficiency and generalization of the model by discarding the redundant features in the input data [6]. Wrapper-based FS provides an easy way to evaluate classifiers using the subset of features with respect to the original features of the dataset, and it also achieves higher classification performance when compared to filter-based methods [7]. The removal of irrelevant features not only enhances the accuracy of classification models but also reduces their complexity when these models are applied in real time for intrusion detection in large-scale network environments [8].

Despite widespread use, existing wrapper-based and metaheuristic FS approaches exhibit notable shortcomings. Algorithms like GA, PSO, ACO, and CSA, and many of their variants, often suffer from premature convergence, causing the search to get stuck in local optima. Their performance worsens as problem dimensionality increases. In addition, many metaheuristics keep control parameters constant during optimization. The wrapper technique is precise but increases computational time, as it must repeatedly score each possible subset against a classifier.

To address these challenges, it is important to apply optimization algorithms that can adapt their search strategies during execution. Dynamic optimization approaches show that algorithms should detect environmental changes and adjust their search behavior by exploiting previous search knowledge to improve convergence performance [9]. In high-dimensional IDS problems involving FS, the search landscape will be rough and Sparse. The approach needs to adjust dynamically to expand the search region when diversity is required and to shift toward intensified exploitation as the algorithm converges toward the optimal solution. The approach will fail when the algorithms employed have constant parameters, which often results in poor performance as data sets grow.

The Dynamic Ant Colony Optimization (DyACO) algorithm is an adaptive extension of feature selection techniques based on the ACO algorithm, which overcomes the limitations of static search mechanisms (i.e., fixed pheromone evaporation rates, predefined heuristic information, and unchanged transition strategies) in high-dimensional IDS environments. Traditional ACO-based FS techniques usually have fixed pheromone evaporation rates, static heuristic information, and stationary transition rules throughout the entire optimization process. In DyACO, among other mechanisms, heuristic information is dynamic, pheromone evaporation rates vary with iterations, pheromone updates are based on reward and punishment

strategies, and state transition rules are adaptive. Such methods help the algorithm achieve a better balance between exploration and exploitation, prevent pheromone trail stagnation, and improve its capability to discover more compact and discriminative feature subsets. The adaptive parameter control strategy allows DyACO to change its search patterns as the optimization process proceeds, making the feature selection framework less static for high-dimensional IDS datasets, where avoiding local optima and selecting informative attributes are crucial.

Although DyACO was originally designed for continuous optimization tasks, adapting it into a binary framework presents new opportunities for IDS FS. Binary DyACO enables ants to construct inclusion–exclusion vectors representing candidate feature subsets, allowing dynamic search mechanisms to operate in a discrete domain. Embedding this binary DyACO within a wrapper architecture further enhances its utility by enabling direct feedback from classification performance during optimization. Such integration leverages both DyACO's adaptive search strengths and the classifier's discriminative capabilities, resulting in a more refined and practical approach to selecting high-impact features for network intrusion detection.

Based on the aforementioned interests, this work proposes a DyACO-based wrapper FS approach, named DyACO. This algorithm aims to optimize IDSs. DyACO stands out for the following reasons: 1) the proposed algorithm uses an adaptive pheromone function, heuristic function, and transition function for the binary variant of DyACO; 2) the algorithm employs the wrapper approach, combining DyACO for the search process with the CART evaluation function and the GRU classifier for the classifier selection process; 3) the algorithm outperforms current methods in terms of the solutions' compactness, speed of convergence, and detection capabilities on the CIC-IDS2018, UNSW-NB15, and NSL-KDD datasets.

II. Literature Review

Shwaysh, et al. [10] introduced a hybrid IDS model that combines information gain with an autoencoder for dimensionality reduction and an ensemble of gradient-boosted classifiers. Zouhri, et al. [11] proposed a comprehensive comparison of univariate and multivariate filter-based FS methods across a wide array of IDS datasets. Barbosa, et al. [12] developed a filtering technique based on graph modeling of Pareto-dominant, informative, and correlated features. This technique discovers informative features using betweenness centrality on a feature-constructed graph, yielding a compact, discriminative subset with high accuracy using only a few features. Habeeb and Babu [13] proposed a two-level FS method that combines coarse correlation filtering with hybridization of the Whale Optimization Algorithm (WOA) and Genetic Algorithm (GA).

Maseno and Wang [14] proposed a sequential wrapper approach that combines an optimized extreme learning machine with SVM classification. Umer, et al. [15] suggested a wrapper-based FS pipeline integrated with a Multi-Head Attention transformer classifier. Anagha, et al. [16] proposed a framework for jointly incorporating SHAP values, correlation analysis, and information gain to select relevant features in deep learning-based IDS classification tasks. Al-Yaseen, et al. [17] proposed a wrapper method that applies Differential Evolution (DE) for FS and Extreme Learning Machine (ELM) for subset evaluation. Halim, et al. [18] suggested an enhanced GA–based FS strategy with tuned parameters and a novel fitness function to preserve essential information while minimizing dimensionality.

Nazir and Khan [19] introduced a wrapper approach using a Tabu Search (TS) algorithm coupled with an RF process to evaluate feature subsets. Pourardebil Khah, et al. [20] proposed a hybrid multi-phase approach that incorporates heuristic filters, fuzzy TOPSIS, and a Discrete Gray Wolf Optimizer (DGWO) for metaheuristic refinement. Hosseini, et al. [21] proposed a hybrid optimizer for OSMOGA that couples the Slime Mold Algorithm (SMA) with GA, using opposition-based learning to improve GA population diversity. This optimizer provides efficient exploration, along with the exploitation strengths of GA and SMA, ensuring accuracy and accelerating learning within GA for the IDS problem. Laamari and Kamel [22] suggested a multi-objective binary bat algorithm within a wrapper FS framework using Decision Trees (DTs) for evaluation.

Although these studies have demonstrated the effectiveness of feature selection and optimization techniques for IDS, their limitations remain evident. Filter-based approaches generally evaluate features independently and may ignore classifier-related dependencies, while wrapper-based methods provide better classification feedback at the cost of higher computational complexity. Furthermore, optimization-based approaches using GA, PSO, DE, and other metaheuristic algorithms can improve feature selection performance; however, they may suffer from premature convergence, limited adaptability, and difficulty in exploring high-dimensional feature spaces. Hybrid optimization strategies can enhance search capability, but they often introduce additional parameters and computational complexity.

III. Background and Preliminaries

A. Gated Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are commonly used to process temporal data; vanilla RNNs may suffer from unstable gradients and loss of long-term dependencies. Gated Recurrent Units (GRUs) were introduced to address the pitfalls of vanilla RNNs, using an innovative gating mechanism to control information flow over time. As shown in Fig. 1, the main difference between GRUs and LSTMs is that GRUs use both an update gate and a reset gate, which reduces the total number of parameters and helps prevent overfitting. GRUs are an excellent candidate for intrusion detection, given the inherent temporal dependencies in network traffic data and the need for efficient real-time processing. A GRU cell receives an input vector x_t at time step t and combines it with the previous internal state h_{t-1}. The first gating component controls how much past information to preserve. Its activation is computed using a sigmoid-based transformation as expressed in Eq. (1).

$$\Upsilon_t^{(u)} = \sigma(W_u[h_{t-1}, x_t]) \quad (1)$$

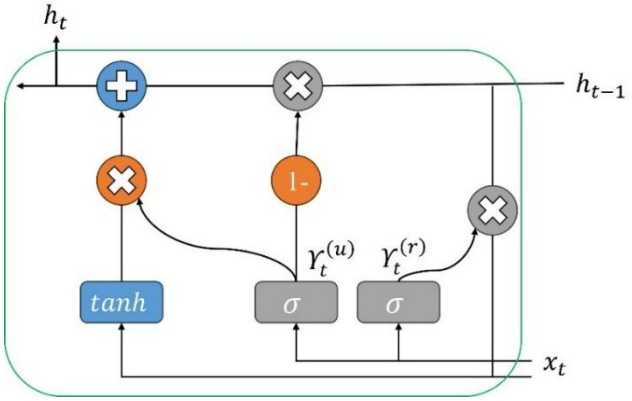


Fig. 1. Internal computational structure of a GRU cell.

The second gate, the reset controller, regulates how much previously forgotten hidden content is used to produce a new candidate state. This mechanism allows the model to selectively ignore irrelevant past information, promoting the sensitivity needed for network attack behaviors with abrupt changes in behavior sequences. Eq. (2) defines the reset gate.

$$r_t^{(r)} = \sigma(W_r[h_{t-1}, x_t]) \tag{2}$$

Using the reset-modulated hidden state, the GRU computes a candidate activation that represents a potential new memory value, incorporating both fresh input and selectively filtered historical data. This candidate is generated via a nonlinear transformation, as expressed in Eq. (3).

$$\widehat{h_t} = tanh\left(W_c\left[r_t^{(r)} \odot h_{t-1}, x_t\right]\right) \tag{3}$$

The final hidden state at a given time step is a function of an interpolation between the preserved old hidden state and the candidate hidden state. This function is modulated by the update gate, which can be viewed as a control gate that enables the cell to maintain a smooth flow of features rather than introduce novelty, as described by Eq. (4).

$$h_t = \left(1 - r_t^{(u)}\right) \odot h_{t-1} + r_t^{(u)} \odot \widehat{h_t} \tag{4}$$

B. Decision Tree Classifier

DTs are hierarchical machine learning algorithms that mimic a human analyst's decision-making process by iteratively dividing a dataset into increasingly homogeneous groups. A DT consists of a root node that is initialized with the dataset, which is later split into branches based on a series of decision statements applied to the values of specific features, one by one. A split within a decision tree occurs on a particular attribute, and the terminal nodes contain the predicted classes, known as leaves.

Several variants of DT algorithms have been developed, including ID3 and C4.5, which employ multiway splits and entropy-based impurity measures. However, these variants can produce overly large structures with deep branching patterns, adding unnecessary complexity. In contrast, the Classification and Regression Tree (CART) methodology uses a binary division strategy, simplifying the tree's architecture and reducing training overhead. CART uses a bottom-up impurity-minimization approach that repeatedly selects the attribute that best separates classes at each node. CART measures split quality using Gini impurity, which assesses the heterogeneity of the class distribution within a subset. For a dataset D consisting of K classes, the impurity is computed using Eq. (5).

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|}\right)^2 \tag{5}$$

where, C_k denotes the collection of samples belonging to class k. A smaller impurity value indicates a more uniform class distribution within the node, thereby signaling a purer partition. When an attribute A is used to divide the dataset into subsets $D_1, D_2, \ldots, D_n$, the overall impurity after splitting is obtained as a weighted combination of the individual impurities, as formulated in Eq. (6).

$$\text{Gini}(D \mid A) = \sum_{i=1}^{n} \left(\frac{|D_i|}{|D|}\right) \text{Gini}(D_i) \tag{6}$$

C. Wrapper Fitness Evaluation

In wrapper-based FS, the search process relies on a scoring mechanism that reflects how well a candidate subset supports classification accuracy while controlling model complexity. Unlike filter approaches, which use statistical proxies, wrapper strategies directly measure a classifier's predictive performance for each subset under evaluation. This lets the optimization algorithm favor feature sets with better real-world detection performance. Because selecting feature sets with very large subset sizes can lead to overfitting and high computational costs, the fitness function must simultaneously weigh predictive accuracy and dimensionality reduction. To formalize this balance, the optimizer assesses each candidate solution using a composite metric that integrates both classification precision and the proportion of selected features. Let P_{cls} represent the accuracy delivered by the classifier (in this case, CART), and let F_{sel} and F_{tot} denote the number of chosen features and the total available attributes, respectively. The resulting fitness score is minimized and is defined in Eq. (7).

$$\text{Fitness} = \min\left[\alpha \cdot \frac{1}{P_{\text{cls}}} + \beta \cdot \frac{F_{\text{sel}}}{F_{\text{tot}}}\right] \tag{7}$$

D. Data Preprocessing Pipeline

Intrusion detection datasets typically contain heterogeneous attributes, missing entries, duplicated flows, and large numerical disparities. If not handled systematically, these irregularities can significantly distort classifier behavior. Therefore, a structured preprocessing workflow is essential to ensure that the learning algorithms receive stable, unbiased, and properly scaled data representations. In this study, a three-stage pipeline comprising data cleansing, categorical encoding, and feature normalization is adopted to prepare all traffic samples for evaluation and optimization.

The first stage focuses on cleaning the raw dataset. Common issues include incomplete records, inconsistent formats, and repeated observations introduced during collection. Eliminating null entries and duplicates ensures that the subsequent optimization process does not overfit to noisy or redundant

patterns. This step is particularly important for intrusion detection since replicated benign flows or malformed attack records can bias the classifier, leading to artificially inflated accuracy or unstable convergence in deep models.

After cleaning, the second stage transforms categorical attributes. Many IDS datasets contain symbolic fields describing protocol categories, service types, connection states, or flag indicators. To incorporate these attributes into numerical learning systems, a translation mechanism is required. In this work, label encoding is applied, assigning a unique integer index to each category.

The third stage addresses feature scaling, a crucial step for stabilizing gradient-based optimization and reducing the influence of attributes with disproportionately large ranges. Large eigenvalue discrepancies can cause erratic cost-surface behavior, producing significant fluctuations during training and slowing model convergence. To mitigate this issue, a min–max normalization scheme is used, mapping each feature value v to a bounded range [0,1]. The transformation is computed according to Eq. (8).

$$v_{norm} = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{8}$$

IV. Proposed Method

The proposed DyACO framework extends classical ACO by integrating dynamic adaptation mechanisms tailored for high-dimensional FS in network intrusion detection. Fig. 2 provides a detailed procedural flow of the DyACO optimization process, beginning with parameter initialization and progressing through iterative solution construction, pheromone updating, and termination. In conventional ACO, pheromone trails, heuristic values, and state-transition rules remain fixed across iterations, often leading to stagnation or premature convergence when searching large binary spaces. To address these limitations, DyACO incorporates four intertwined dynamic mechanisms: adaptive heuristics, iteration-aware pheromone volatility, reward–penalty reinforcement, and parameter-adjusted state transitions.

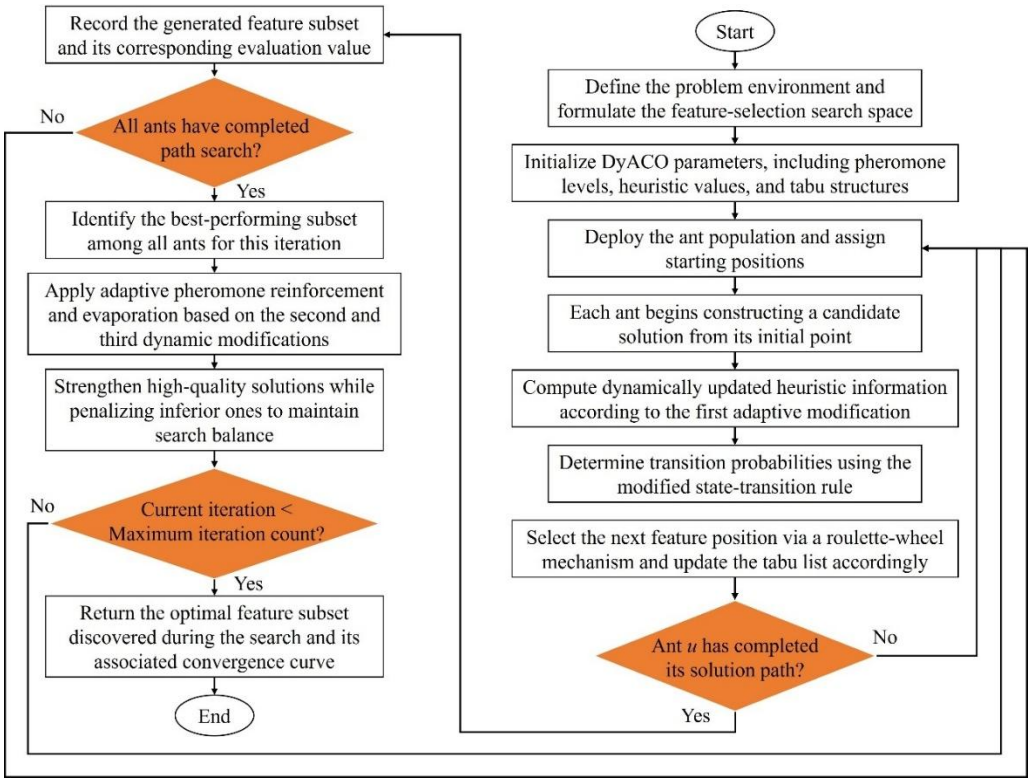


Fig. 2. Flowchart of the DyACO framework.

A. Mathematical Model of Traditional ACO

Traditional ACO simulates cooperative behavior observed in real ant colonies. Artificial ants traverse a solution graph, a graph where selecting or discarding each feature corresponds to choosing one of two nodes, while depositing virtual pheromones that bias future ant decisions. This behavior is characterized by two fundamental processes: the transition probability, which determines how ants choose their next binary decision, and pheromone revision, which regulates the colony's memory. When constructing a feature subset, each ant begins with an empty solution and progressively decides whether to include or exclude each feature. For a given ant u at decision point f, the chance of selecting option o∈{0,1} (exclude/include) is determined by the classical transition model denoted by Eq. (9).

$$\prod_{f,o}^{u} = \begin{cases} \frac{\Lambda_{f,o}(t)^{\alpha_0} \cdot \Gamma_{f,o}(t)^{\beta_0}}{\sum_{o' \in \{0,1\}} \Lambda_{f,o'}(t)^{\alpha_0} \cdot \Gamma_{f,o'}(t)^{\beta_0}} & \text{if } o \text{ is feasible} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where, $\Lambda_{f,o}(t)$ is the pheromone intensity for choosing option o for feature f during iteration t, $\Gamma_{f,o}(t)$ is the heuristic

bias for option o, α_0 and β_0 are weight pheromone memory and heuristic guidance, respectively. This formulation governs initial exploration but is insufficient for high-dimensional intrusion datasets because static parameters cannot adapt to evolving search conditions. The pheromone values are updated after each iteration to reinforce desirable decisions. Classical evaporation and deposition follow Eq. (10).

$$\Lambda_{f,o}(t+1) = (1-\delta_0)\Lambda_{f,o}(t) + \sum_{u=1}^{U} \Delta\Lambda_{f,o}^{u}(t) \quad (10)$$

The deposit amount for ant u, is determined by Eq. (11).

$$\Delta\Lambda_{f,o}^{u}(t) = \begin{cases} \frac{C}{L_u} & \text{if ant u selected o for } f \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where, L_u is the solution quality (e.g., inverse fitness), C is a constant reward, and δ_0 is the fixed evaporation rate. In complex FS scenarios, fixed evaporation and simple reinforcement can cause stagnation due to over-concentration of pheromones on early-discovered subsets.

To overcome the previously described limitations, DyACO introduces adaptive mechanisms that evolve over the course of the search. Rather than treating each decision as static, DyACO continuously adjusts heuristics, pheromone decay, reinforcement, and transition probabilities over iterations.

In classical ACO, heuristic information is often based on simple metrics (e.g., inverse distance). In FS, the "heuristic" should instead reflect a feature's usefulness. DyACO introduces a dynamic weighting function using a Gaussian-shaped modulation that varies with iteration count. The dynamic heuristic is defined in Eq. (12).

$$\Gamma_{f,o}(t) = \Psi_t \cdot \frac{X}{d_{f,o}^2} \quad (12)$$

The temporal scaling factor follows a modified normal distribution, expressed by Eq. (13).

$$\Psi_t = \exp\left(-3\left(\frac{t}{T}\right)^2\right) \quad (13)$$

where, t is the current iteration, T is the maximum number of iterations, $d_{f,o}$ is a domain-specific heuristic score (e.g., relevance measure), and X is a constant. This mechanism ensures strong heuristic guidance early on while diminishing its effect later, promoting better global exploration. Traditional ACO uses a fixed pheromone evaporation parameter, which is unsuitable for high-dimensional FS because it either over-strengthens or over-weakens pheromone memory. DyACO adopts a sigmoid-based dynamic evaporation rate, allowing the pheromone memory to adapt as the search progresses. The dynamic pheromone volatility is computed according to Eq. (14).

$$\delta_t = \frac{1}{1 + e^{-\left(A\cdot\frac{T-t}{T} + B\cdot\frac{t}{T}\right)}} \quad (14)$$

A and B control the transition shape. Early iterations maintain lower evaporation (thus promoting exploitation), while later iterations increase evaporation to encourage diversification and overcome stagnation. Classical reinforcement accumulates pheromone only on good solutions. DyACO modifies this by rewarding strong feature subsets and penalizing the worst ones, encouraging the colony to abandon inferior search regions. The updated pheromone model is expressed in Eq. (15).

$$\Lambda_{f,o}(t+1) = (1-\delta_t)\Lambda_{f,o}(t) + \sum_{u=1}^{U} \Delta\Lambda_{f,o}^{u}(t) \quad (15)$$

The deposit term is adjusted as stated in Eq. (16).

$$\Delta\Lambda_{f,o}^{u}(t) = \begin{cases} \frac{3C}{L_u} + \mu \cdot \frac{L_{best} - L_u}{L_{avg} - L_{best}}, & L_u \le L_{best} \\ \frac{C}{L_u} + \varepsilon \cdot \frac{L_u - L_{worst}}{L_{worst} - L_{avg}}, & L_u \ge L_{best} \\ \frac{C}{L_u}, & \text{otherwise} \end{cases} \quad (16)$$

where, L_{best}, L_{worst}, and L_{avg} are iteration-level statistics, and μ and ε are reinforcement weights. This formulation accelerates convergence while avoiding the amplification of poor decisions. To balance exploration and exploitation, DyACO introduces two mechanisms, namely, the feature-smoothness heuristic and the updated transition rule. A turning-angle heuristic from path planning is reinterpreted here as subset smoothness, discouraging abrupt changes in FS patterns that may lead to unstable learning performance. Its scaling factor is computed using Eq. (17).

$$\Theta_t = \frac{1}{e^{(\omega_t/2)}} \quad (17)$$

where, ω_t quantifies inconsistency in consecutive selections in the current ant's partial feature subset. Eq. (18) presents the improved transition probability combining pheromone, heuristic, and subset-smoothness.

$$\prod_{f,o}^{u}(t) = \begin{cases} \frac{\Lambda_{f,o}(t)^{\alpha_1} \cdot \Gamma_{f,o}(t)^{\beta_1} \cdot \Theta_t^{\gamma_1}}{\sum_{o' \in \{0,1\}} \Lambda_{f,o'}(t)^{\alpha_1} \cdot \Gamma_{f,o'}(t)^{\beta_1} \cdot \Theta_t^{\gamma_1}}, & o \in \{0,1\} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

A dynamically decaying exploitation parameter q_t is adopted, computed according to Eq. (19).

$$q_t = q^* \cdot \exp\left(-0.5\left(\frac{t}{T}\right)^2\right) \quad (19)$$

B. Training and Evaluation of Classifiers

As a result of the proposed FS procedure, a classification model can be trained to distinguish malicious traffic from legitimate network traffic using the selected features. In this work, a GRU–based neural architecture serves as the primary classifier because it efficiently learns temporal dependencies in flow-based intrusion data. After feature selection, the selected attributes of each network flow are organized by temporal occurrence to construct sequential input representations for the

GRU model. Each traffic sample is therefore represented as a feature sequence, allowing the GRU network to capture temporal dependencies and behavioral patterns associated with different network activities. The trained model is then validated on an independent test partition to assess its generalization ability.

To ensure a fair comparison across competing FS strategies, the same GRU configuration is used throughout the experiments. The implemented intrusion detection network consists of five hidden layers constructed around GRU cells, forming a deep sequential learning pipeline tailored for network traffic analysis. Table I summarizes the architectural specifications, including the number of hidden units, selected feature dimensionality (#), and output class cardinality (*) for each dataset.

TABLE I. HYPERPARAMETER CONFIGURATION OF THE IMPLEMENTED GRU-BASED IDS MODEL

Component	Specification	Value
Input layer	Feature dimension	#
Dense Layer 1	Linear neurons	100
Dense Layer 2	Linear neurons	64
Recurrent Layer 1	GRU cells	256
Recurrent Layer 2	GRU cells	128
Dense Layer 3	Linear neurons	100
Output layer	Linear units (classification)	*
Training batch size	–	1024
Total epochs	–	300
Dropout probability	–	0.5
Learning rate	–	0.0003
Optimization algorithm	–	Adam (Neural Network variant)
Activation function	–	ReLU
Loss function	–	Cross-Entropy

V. RESULTS AND DISCUSSION

This section reports the empirical evaluation of the proposed DyACO framework across two complementary experimental dimensions. First, DyACO is benchmarked using widely adopted optimization test functions from the CEC suite to validate its global search ability and robustness in heterogeneous landscapes. Then, DyACO is assessed within the intrusion detection domain, examining three representative datasets, including CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD, where its FS efficiency and classification performance are compared against several well-established metaheuristic baselines.

To comprehensively evaluate DyACO, three publicly available benchmark datasets were employed, including CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD. Collectively, these databases span both legacy and modern network environments, include diverse attack families, and exhibit variations in feature dimensionality and class imbalance, conditions that challenge both FS algorithms and IDS classifiers. The NSL-KDD dataset refines the well-known KDDCup99 dataset by removing redundant and duplicate traffic flows, thereby minimizing training bias and mitigating the dominance of majority attack classes. NSL-KDD contains four attack types and normal traffic instances and provides 125,973 records for training and 22,543 for testing. UNSW-NB15 was generated using the IXIA PerfectStorm testbed to emulate realistic contemporary cyber threats.

CSE-CIC-IDS2018 is an enterprise-grade repository developed by the Canadian Network Security Institute that models realistic operational environments. It includes normal activity, along with a range of modern cyberattacks such as brute-force attacks, infiltration, botnet behavior, Web exploits, and distributed Denial-of-Service (DDoS). Given its high dimensionality and complex traffic dynamics, CSE-CIC-IDS2018 is a particularly rigorous testbed for evaluating advanced FS systems like DyACO. As summarized in Table II, the dataset comprises benign traffic alongside diverse contemporary attack categories.

TABLE II. DISTRIBUTION OF ATTACK CATEGORIES AND FLOW VOLUMES IN THE CSE-CIC-IDS2018 DATASET

Attack category	Example attack instances	Number of flows
Normal traffic	Benign	13,484,708
Web-based exploits	Brute Force-Web, Brute Force-XSS, SQL Injection	611; 230; 87
DoS incidents	SlowHTTPTest, GoldenEye, Slowloris, Hulk	139,890; 41,508; 10,990; 461,192
Infiltration threats	Infiltration	161,934
Brute-force attempts	FTP-BruteForce, SSH-Bruteforce	193,360; 187,589
Botnet activity	Bot	286,191
DDoS-related attacks	HOIC, LOIC-HTTP, LOIC-UDP	686,012; 576,191; 1,730
Overall total	–	16,232,223

In this study, the performance of the GRU-based intrusion classifier, fed with the feature subsets recommended by DyACO, is quantified using a set of well-established metrics derived from the confusion matrix. The confusion matrix summarizes the model's predictions by categorizing samples into four groups: True Positives (TPs) representing correctly detected attacks, True Negatives (TNs) for correctly identified normal traffic, False Positives (FPs) referring to benign samples incorrectly flagged as malicious, and False Negatives (FNs) denoting attacks that the classifier failed to identify. To comprehensively evaluate the classifier, four primary measures are adopted: accuracy, recall, precision, and F1-score. Accuracy reflects the overall proportion of correctly classified samples and serves as a broad indicator of predictive performance, computed using Eq. (20).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

To better capture the algorithm's ability to identify malicious behavior, recall [(Eq. (21)] is employed. Recall quantifies the fraction of actual attacks the algorithm detects.

$$Recall = \frac{TP}{TP + FN} \quad (21)$$

Complementing recall, precision measures the reliability of positive predictions by evaluating how many detected attacks are indeed malicious. It is calculated according to Eq. (22).

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

Since high recall with low precision (or vice versa) is undesirable, the F1-score provides a harmonic mean of these two metrics, yielding a balanced indicator of IDS performance. Eq. (23) defines the F1-score.

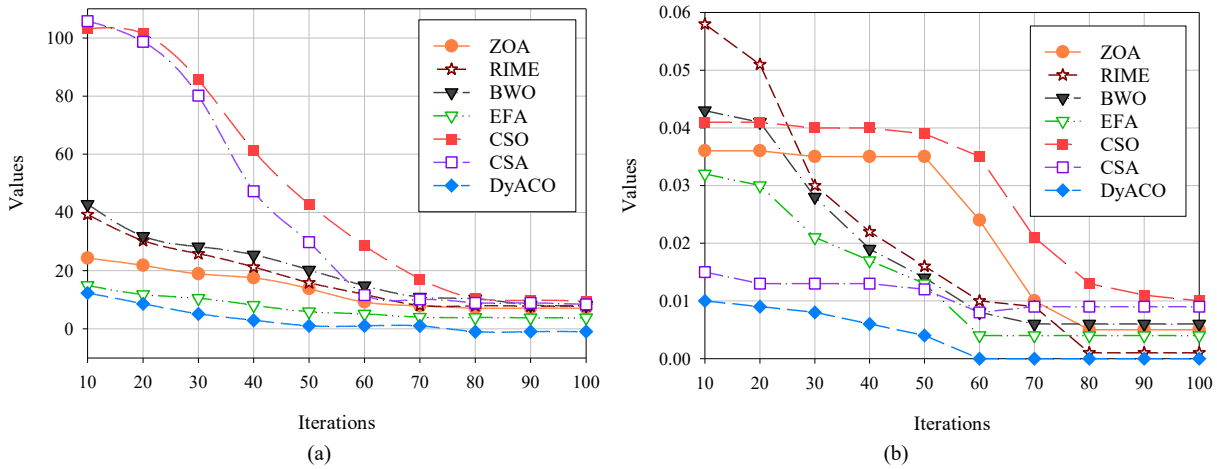
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (23)$$

To establish the optimization capability of DyACO, the framework was initially benchmarked using a collection of well-known continuous test functions widely adopted in CEC competitions. As summarized in Table III, these functions span diverse search landscapes, varying in modality, curvature, and search-space boundaries. DyACO was evaluated against six competitive baseline optimizers, including Zebra Optimization Algorithm (ZOA) [23], Rime Optimization Algorithm (RIME) [24], Beluga Whale Optimization (BWO) [25], Enhanced Fireworks Algorithm (EFA) [26], Chicken Swarm Optimization (CSO) [27], and Crow Search Algorithm (CSA) [28]. All experiments used uniform settings, including a population size of N=60, a 30-dimensional search space, and a maximum of 100 iterations.

TABLE III. BENCHMARK OPTIMIZATION FUNCTIONS USED IN THIS STUDY, ALONG WITH THEIR INPUT DOMAINS AND GLOBAL SOLUTIONS

Benchmark	Reformulated expression	Search interval	Optimal value
Rosenbrock	$g(\mathrm{u}) = \sum_{j=1}^{m-1} [100(u_{j+1} - u_j^2)^2 + (1 - u_j)^2]$	$[-5, 10]$	0
Schaffer	$g(v_1, v_2) = 0.5 + \frac{\sin^2(v_1^2 - v_2^2) - 0.5}{[1 + 0.001(v_1^2 + v_2^2)]^2}$	$[-100, 100]$	0
Booth	$g(v_1, v_2) = (v_1 + 2v_2 - 7)^2 + (2v_1 + v_2 - 5)^2$	$[-10, 10]$	0
Easom	$g(v_1, v_2) = -\cos(v_1)\cos(v_2)\exp(-[(v_1 - \pi)^2 + (v_2 - \pi)^2])$	$[-100, 100]$	–1
Rastrigin	$g(\mathrm{u}) = 10m + \sum_{j=1}^{m} [u_j^2 - 10\cos(2\pi u_j)]$	$[-5.12, 5.12]$	0
Ackley	$g(\mathrm{u}) = -20\exp(-0.2\sqrt{\frac{1}{m}\sum_{j=1}^{m} u_j^2}) - \exp(\frac{1}{m}\sum_{j=1}^{m}\cos(2\pi u_j)) + 20 + e$	$[-5, 5]$	0

Fig. 3 visually presents convergence behavior and final optimization performance, highlighting the superiority of the proposed method. The intrusion detection experiments on CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD collectively demonstrate DyACO's effectiveness in improving FS quality and classification performance across heterogeneous network environments. DyACO consistently outperforms baseline metaheuristics, demonstrating its ability to navigate high-dimensional search spaces while maintaining stable convergence. The visual comparisons in Fig. 4–7 further reinforce these findings by clearly separating DyACO from competing strategies across accuracy, precision, recall, and F1-score distributions.



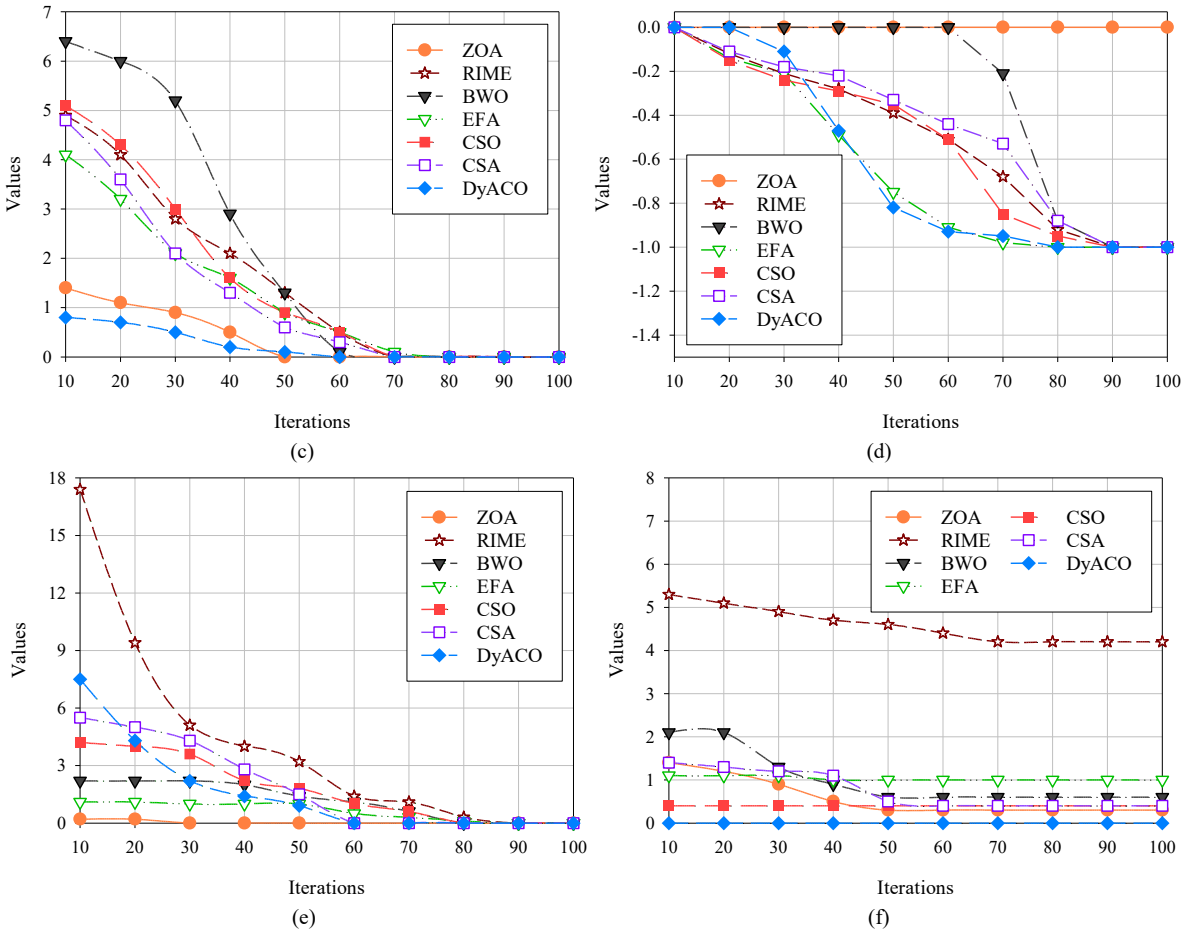


Fig. 3. Convergence curves comparing DyACO with baseline optimizers across benchmark test functions: Rosenbrock (a), Schaffer (b), Booth (c), Easom (d), Rastrigin (e), Ackley (f).

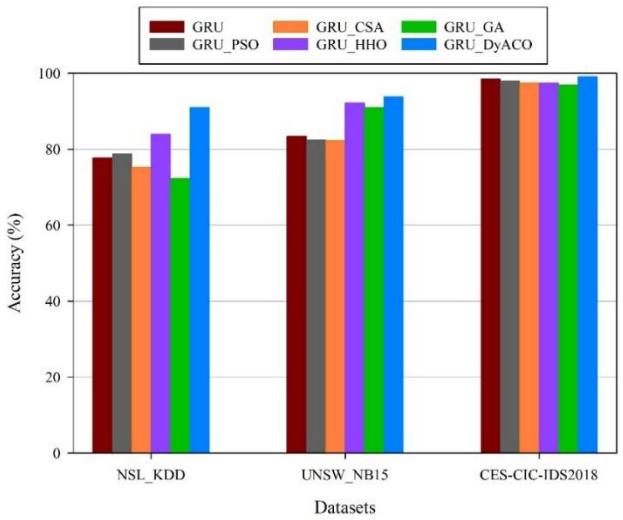


Fig. 4. Accuracy comparison of GRU- driven IDS models across benchmark datasets.

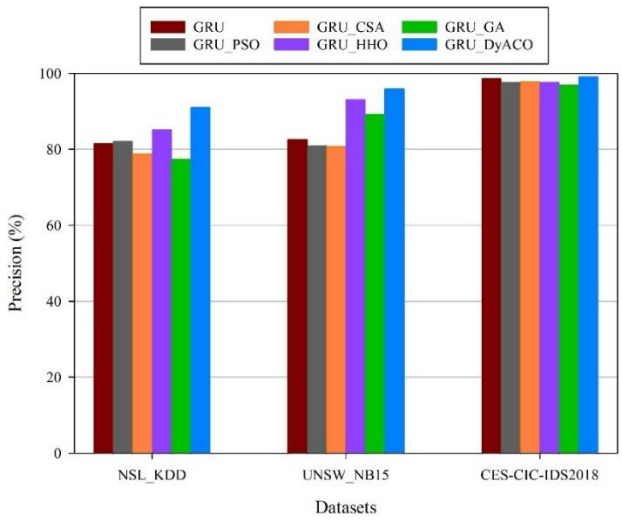


Fig. 5. Precision comparison of GRU- driven IDS models across benchmark datasets.

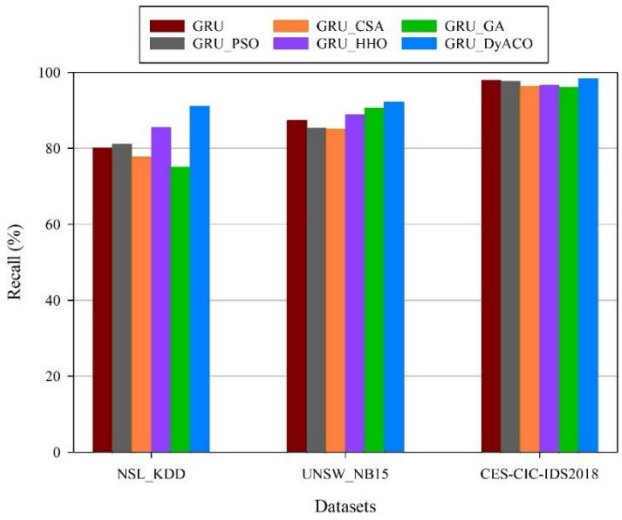


Fig. 6. Recall comparison of GRU- driven IDS models across benchmark datasets.

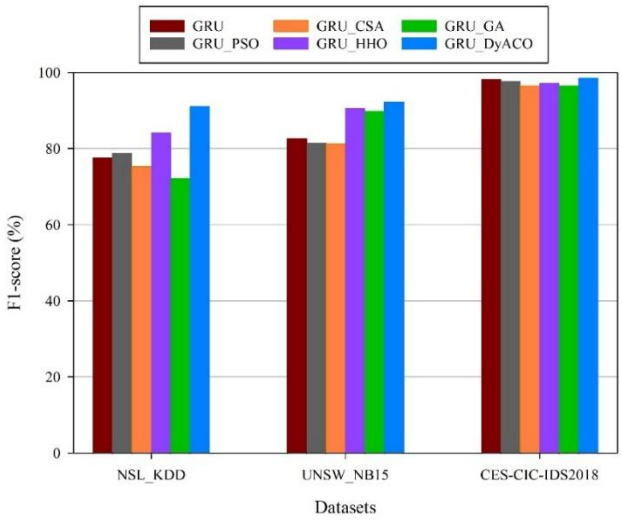


Fig. 7. F1-score comparison of GRU- driven IDS models across benchmark datasets.

VI. Conclusion

In this study, a dynamic wrapper-based method called DyACO, based on the ACO algorithm, is proposed to efficiently address high-dimensional FS problems for NID. By introducing adaptive heuristics, iteration-dependent pheromone evaporation rates, magnitude-based reward–penalty functions in the reinforcement scheme, and adaptive state transition rules, the DyACO algorithm overcomes the limitations of conventional metaheuristic algorithms. Experimental evaluations on the CSE-CIC-IDS2018, UNSW-NB15, and NSL-KDD datasets demonstrate that DyACO can obtain compact subsets of discriminative features, reduce feature dimensionality, and achieve high detection performance. Based on accuracy, precision, recall, and F1-score, DyACO demonstrates superior performance compared with the considered optimization algorithms while maintaining the computational efficiency required for wrapper-based methods.

References

[1] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019, doi: https://doi.org/10.1109/JIOT.2019.2933518.

[2] C. Hamroun, A. Fladenmuller, M. Pariente, and G. Pujolle, "Intrusion detection in 5G and Wi-Fi networks: A survey of current methods, challenges & perspectives," IEEE Access, 2025.

[3] E. Rivandi, "Application of Machine Learning Techniques to Fraud Detection in Financial Transactions," Journal of Intelligent Decision Making and Granular Computing, vol. 2, no. 1, pp. 288-303, 2026, doi: https://doi.org/10.31181/jidmgc21202647.

[4] D. Manivannan, "Recent endeavors in machine learning-powered intrusion detection systems for the internet of things," Journal of Network and Computer Applications, vol. 229, p. 103925, 2024.

[5] V. Hayyolalam, B. Pourghebleh, and A. A. Pourhaji Kazem, "Trust management of services (TMoS): investigating the current mechanisms," Transactions on Emerging Telecommunications Technologies, vol. 31, no. 10, p. e4063, 2020, doi: https://doi.org/10.1002/ett.4063.

[6] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," Computers & Security, vol. 137, p. 103587, 2024.

[7] E. Zorarpaci, "A fast intrusion detection system based on swift wrapper feature selection and speedy ensemble classifier," Engineering Applications of Artificial Intelligence, vol. 133, p. 108162, 2024.

[8] J. Maldonado and M. C. Riff, "An evolutionary wrapper to support intrusion detection system configuration," Computers & Security, p. 104478, 2025.

[9] A. Birashk, J. K. Kordestani, and M. R. Meybodi, "Cellular teaching-learning-based optimization approach for dynamic multi-objective problems," Knowledge-Based Systems, vol. 141, pp. 148-177, 2018, doi: https://doi.org/10.1016/j.knosys.2017.11.016.

[10] M. M. Shwaysh et al., "Adaptive hybrid information gain and autoencoder-based feature selection with ensemble recurrent extreme learning machine for enhanced network intrusion detection systems," Journal of Network and Systems Management, vol. 34, no. 1, pp. 1-30, 2026.

[11] H. Zouhri, A. Idri, and A. Ratnani, "Evaluating the impact of filter-based feature selection in intrusion detection systems," International Journal of Information Security, vol. 23, no. 2, pp. 759-785, 2024.

[12] G. N. N. Barbosa, M. Andreoni, and D. M. F. Mattos, "Optimizing feature selection in intrusion detection systems: Pareto dominance set approaches with mutual information and linear correlation," Ad Hoc Networks, vol. 159, p. 103485, 2024.

[13] M. S. Habeeb and T. R. Babu, "Coarse and fine feature selection for network intrusion detection systems (IDS) in IoT networks," Transactions on Emerging Telecommunications Technologies, vol. 35, no. 4, p. e4961, 2024.

[14] E. M. Maseno and Z. Wang, "Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection," Journal of big data, vol. 11, no. 1, p. 24, 2024.

[15] M. Umer, M. Tahir, M. Sardaraz, M. Sharif, H. Elmannai, and A. D. Algarni, "Network intrusion detection model using wrapper based feature selection and multi head attention transformers," Scientific reports, vol. 15, no. 1, p. 28718, 2025.

[16] A. Anagha, C. Thomas, and N. Balakrishnan, "Optimized intrusion predictions through feature selection methods," Computers & Security, p. 104541, 2025.

[17] W. L. Al-Yaseen, A. K. Idrees, and F. H. Almasoudy, "Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system," Pattern Recognition, vol. 132, p. 108912, 2022.

[18] Z. Halim et al., "An effective genetic algorithm-based feature selection method for intrusion detection systems," Computers & Security, vol. 110, p. 102448, 2021.

[19] A. Nazir and R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," Computers & Security, vol. 102, p. 102164, 2021.

[20] Y. Pourardebil Khah, M. Hosseini Shirvani, and H. Motameni, "A hybrid machine learning approach for feature selection in designing intrusion detection systems (IDS) model for distributed computing networks," The Journal of Supercomputing, vol. 81, no. 1, p. 254, 2025.

[21] S. Hosseini, M. Khorashadizade, and M. Jouyban, "A Hybrid Method Using Slime Mold Algorithm and Genetic Algorithm for Feature Selection Problems in Intrusion Detection Systems," Engineering Reports, vol. 7, no. 7, p. e70254, 2025.

[22] M. A. Laamari and N. Kamel, "A New Multi-Objective Binary Bat Algorithm for Feature Selection in Intrusion Detection Systems," Concurrency and Computation: Practice and Experience, vol. 37, no. 4-5, p. e70000, 2025.

[23] E. Trojovská, M. Dehghani, and P. Trojovský, "Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm," Ieee Access, vol. 10, pp. 49445-49473, 2022.

[24] H. Su et al., "RIME: A physics-based optimization," Neurocomputing, vol. 532, pp. 183-214, 2023.

[25] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm," Knowledge-based systems, vol. 251, p. 109215, 2022.

[26] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in 2013 IEEE congress on evolutionary computation, 2013: IEEE, pp. 2069-2077.

[27] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: chicken swarm optimization," in International conference in swarm intelligence, 2014: Springer, pp. 86-94.

[28] A. G. Hussien et al., "Crow search algorithm: theory, recent advances, and applications," Ieee Access, vol. 8, pp. 173548-173565, 2020.

Data-Driven Modeling and Optimization of Wind Turbine Performance

(Evidence from a Wind Farm in Kosovo)

Naim Baftiu[1], Ana Atanasova[2], Enis Baftiu[3], Tatjana A. Pachemska[4],
Egzon Baftiu[5], Vladimir Zdraveski[6], Petre Lameski[7]
Faculty of Computer Science, Goce Delchev University, Stip, North Macedonia[1, 4]
Faculty of Computer Science and Engineering, "Ss. Cyril and Methodius" University in Skopje, North Macedonia[2, 6, 7]
IT Director, Northern Technologies International Corporation (NTIC), Circle Pines, Minnesota, United States of America[3]
Software Developer, Adaptech LLC, Austrabe 21, 65527 Niedernhausen, Germany[5]

***Abstract*—This study proposes an integrated data-driven framework that couples wind power forecasting with the operational optimization of wind turbines under real meteorological conditions. Unlike conventional approaches that focus primarily on improving the predictive accuracy of energy production models, the proposed framework extends forecasting toward proactive operational decision-making, establishing a direct link between prediction, turbine operation, and energy output. Machine learning techniques are used to model the complex nonlinear relationships between meteorological variables and turbine power output, after which a grid-based optimization procedure identifies the operating conditions that maximize generation. The framework is evaluated on ten years of meteorological data for Kosovo covering the period 2015–2024, providing an empirical basis for assessing its applicability. Particular attention is given to identifying the most influential operational and environmental parameters and to determining optimal operating conditions under varying wind regimes. Feature-importance analysis indicates that wind speed and wind direction dominate the response of the model, while rotor speed and blade pitch contribute almost as strongly as the external conditions. Applying the optimization layer to the predicted operating points yields an average increase in power output of 73.70 kW, corresponding to an estimated annual energy gain of approximately 645.60 MWh. A comparative evaluation further shows that a recurrent architecture based on Long Short-Term Memory achieves substantially lower forecasting error than the Random Forest model employed for optimization. The main contribution of the study is a unified forecasting–optimization framework that moves beyond predictive accuracy toward measurable performance improvement, and that can serve as a practical decision-support tool for wind farm operators.**

***Keywords*—Wind energy; wind power forecasting; machine learning; operational optimization; wind turbine performance; random forest; long short-term memory**

I. Introduction

The integration of wind energy into modern power systems is constrained less by the availability of the resource than by its variability. Because the power delivered by a turbine responds nonlinearly to a set of interacting meteorological variables, system operators require reliable estimates of production in order to schedule generation, balance the network, and limit curtailment. A substantial body of work has therefore been devoted to wind power forecasting, ranging from classical statistical models to modern machine learning and deep learning architectures [1]–[4].

The dominant evaluation paradigm in this literature is statistical: competing models are ranked according to how closely their predictions match observed production, using error measures such as the mean absolute error, the root mean squared error and the coefficient of determination. This paradigm has produced considerable methodological progress, but it leaves an important question unanswered. A forecast, however accurate, does not by itself increase the energy delivered by a turbine; it acquires value only when it is translated into an operating decision. Comparatively little attention has been paid to the step that connects the two, namely the use of a trained predictive model as a surrogate over which turbine control parameters can be searched.

The present study addresses this gap by developing a unified forecasting–optimization framework and evaluating it on real meteorological data for Kosovo covering the period 2015–2024. A Random Forest regressor is first trained to predict power output from meteorological and operational variables. The trained model is then used in a second phase as an objective function: for every instance in the test set, combinations of rotor speed and blade pitch are evaluated and the combination that maximizes predicted output is retained. The difference between the power predicted under the current control setting and the power predicted under the optimized setting is reported both instantaneously, in kW, and as an estimated annual energy gain, in MWh.

A. Research Objectives

To achieve this aim, the following objectives were defined:

- To assemble and prepare real meteorological data for Kosovo for the period 2015–2024;
- To develop an accurate model for forecasting power output;
- To optimize the key turbine control parameters, specifically rotor speed and blade pitch;
- To quantify the resulting energy gains in kW and in MWh per year;

- To compare the results obtained with reference studies on wind power forecasting.

B. Contribution of this Study

The contribution of the work is threefold. First, it formulates operational optimization as a search performed directly over a trained forecasting model, so that prediction and control are treated within a single computational pipeline rather than as separate exercises. Second, it quantifies the outcome of that search in physical rather than purely statistical units, expressing the benefit as an average power increment and an estimated annual energy gain. Third, it evaluates the framework on a ten-year meteorological record for Kosovo, a setting characterized by light to moderate wind conditions in which precise turbine control is particularly consequential.

C. Organization of the Study

The remainder of the study is organized as follows. Section II reviews related work on statistical, machine learning, and deep learning forecasting models and on optimization in wind energy systems. Section III describes the data, the construction of the model variables and the proposed forecasting–optimization methodology. Section IV reports the experimental results. Section V discusses their interpretation, implications, and limitations, and Section VI concludes.

II. Related Work

A. Statistical Forecasting Models

Classical time-series approaches remain a standard benchmark in wind power forecasting. The Box–Jenkins family of autoregressive integrated moving average models, including its seasonal variant, provides a well-established framework for representing linear temporal dependence [2], while general methodological guidance for the construction and evaluation of such models is given by Hyndman and Athanasopoulos [3]. Decomposition-based procedures such as Prophet, which describe a series through additive trend, seasonality, and holiday components, have been proposed as robust and easily configurable alternatives [4]. Because these formulations assume approximately linear relationships, their accuracy tends to degrade when the underlying atmospheric processes are strongly nonlinear. The importance of rigorous out-of-sample evaluation in this setting has long been emphasised [16], and broader reviews situate deterministic forecasting within the wider move toward probabilistic energy forecasting [1].

B. Machine Learning Approaches

Ensemble methods based on decision trees are widely used for wind power prediction because they capture nonlinear interactions without requiring an explicit functional form. The Random Forest algorithm aggregates a large number of decorrelated trees and provides, as a by-product, a measure of the relative importance of each input variable [5]. Wei et al. applied this approach to SCADA records for short-term wind power forecasting and reported competitive accuracy relative to conventional baselines [6]. Kernel-based regression constitutes a second established family: support vector regression [7], [8] performs well on moderately sized datasets.

Although its accuracy depends strongly on the selection of the regularisation and kernel parameters, practical guidance has been developed [9].

C. Deep Learning Approaches

Recurrent neural networks with gated memory cells were introduced to address the vanishing-gradient problem that limits the modelling of long-range temporal dependence [10], and a systematic evaluation of the architectural variants of this family is provided by Greff et al. [11]. Applications to wind power are numerous: Liu et al. demonstrated that a Long Short-Term Memory network outperforms conventional statistical baselines on short-horizon forecasting [12], while hybrid formulations that combine a linear model with a recurrent network have been proposed to capture linear and nonlinear structure simultaneously [13]. Convolutional recurrent architectures extend the same principle to spatio-temporal fields [14].

D. Optimization in Wind Energy Systems

Optimization has traditionally been applied in wind energy at the design stage, for instance in the placement of turbines so as to maximize the energy captured across a site [15]. Operational optimization at the level of an individual turbine, in which control parameters are adjusted in response to predicted conditions, has received considerably less attention in the data-driven literature. The present study is positioned at this intersection: it retains the forecasting model as the analytical core, but treats it as a surrogate over which an operating decision is searched, thereby linking the two strands of research described above.

III. Materials and Methods

A. Study Area and Data Source

The meteorological data used in this study were obtained from the renewables.ninja platform for the territory of the Republic of Kosovo and cover the ten years from 2015 to 2024. The retrieved variables comprise wind speed, temperature, humidity, and air density. Fig. 1 presents the statistical distribution of wind speed over the observation period. The distribution follows the Weibull form that is characteristic of wind energy studies, with a high concentration of observations at low and moderate speeds. The modal wind speed lies between approximately 0.8 and 1.2 m/s, and although the majority of measurements fall below 4 m/s, the distribution exhibits a long right tail extending to between 6 and 8 m/s. It is this tail that accounts for the greater part of the energy that can be extracted, since the available power scales with the cube of wind speed. The high absolute frequencies, reaching approximately 16,000 observations in the modal bin, indicate that the models were trained on a sufficiently large sample.

The wind regime shown in Fig. 1 has a direct bearing on the design of the study. Under light to moderate conditions, the operating point of a turbine lies for much of the time on the steep part of the power curve, where small changes in blade pitch and rotor speed translate into comparatively large changes in output. This is precisely the regime in which an optimization layer is expected to be most valuable.

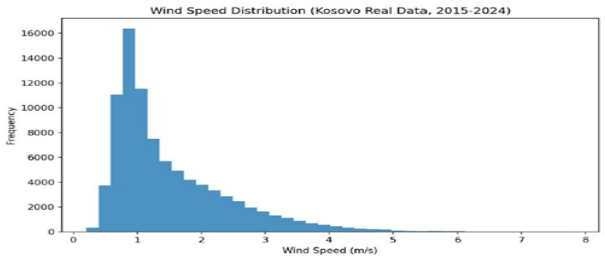


Fig. 1. Statistical distribution of wind speed in Kosovo over the period 2015–2024.

B. Data Preparation

The raw records were harmonised on a common timestamp, filtered to the period 2015–2024, and arranged in time-series format. This procedure ensures the quality and internal consistency of the dataset and provides a uniform temporal resolution for the subsequent modelling stages.

C. Construction of the Model Variables

The variables of the final dataset are summarised in Table I. The pressure variable was not measured directly but derived from the ideal gas law, which links temperature and air density. Because complete SCADA records were not available for the site, the power output, rotor speed and blade pitch variables were modelled so as to reproduce the characteristic operating behaviour of a wind turbine. This is an acknowledged constraint of the present work and is discussed in Section V-D.

TABLE I. VARIABLES OF THE FINAL MODELLING DATASET

Variable	Description	Role
Timestamp	Time index of the observation	Index
WindSpeed	Horizontal wind speed	Predictor
WindDirection	Direction of the incident flow	Predictor
Temperature	Ambient air temperature	Predictor
Humidity	Relative air humidity	Predictor
Pressure	Derived from the ideal gas law	Predictor
RotorSpeed	Rotational speed of the rotor	Control
BladePitch	Pitch angle of the blades	Control
TurbineStatus	Operating state of the turbine	Auxiliary
PowerOutput	Electrical power delivered	Target

D. Train–Test Partition

The dataset was partitioned chronologically, with random shuffling explicitly disabled. Preserving the temporal ordering of the observations is essential in a time-series setting: a random partition would allow information from later periods to influence the fitted model and would therefore produce optimistically biased estimates of out-of-sample accuracy through data leakage.

E. Phase One: the Forecasting Model

The first phase of the framework trains a Random Forest regressor to predict power output from the meteorological and operational variables listed in Table I. The algorithm constructs an ensemble of decision trees and aggregates their individual predictions, which yields three properties that are relevant here: it captures nonlinear relationships without an explicit functional specification; it is robust to noise in the input data; and it provides an interpretable ranking of variable importance [5].

F. Phase Two: Grid-Based Operational Optimization

The second phase reuses the trained model as an objective function. Let $\hat{f}$ denote the fitted regressor, x_t the vector of meteorological conditions observed at time t, and ω and β the rotor speed and blade pitch, respectively. For each instance of the test set, a systematic search is performed over a discrete grid of admissible control values, and the combination that maximises the predicted output is retained:

$$(\omega^*, \beta^*) = \arg\max \hat{f}(x_t, \omega, \beta) \quad (1)$$

The instantaneous benefit of the optimization at time t is the difference between the power predicted under the optimized setting and the power predicted under the control setting actually applied:

$$\Delta P_t = \hat{f}(x_t, \omega^*, \beta^*) - \hat{f}(x_t, \omega_t, \beta_t) \quad (2)$$

The corresponding annual energy benefit is obtained by extrapolating the mean gain over the hours of a full year:

$$E = \Delta\bar{P} \times 8760 / 1000 \quad (3)$$

where, E is expressed in MWh per year and $\Delta\bar{P}$ in kW. Grid search was preferred over gradient-based alternatives because the control space is low-dimensional and bounded, which makes an exhaustive evaluation computationally inexpensive while guaranteeing that the returned solution is globally optimal on the grid. This step transforms the model from a forecasting instrument into a decision-support instrument.

G. Evaluation Metrics

The predictive accuracy of the models was assessed using three standard measures. The mean absolute error reports the average magnitude of the deviations:

$$MAE = (1/n)\ \Sigma\ |y_i - \hat{y}_i| \quad (4)$$

The root mean squared error penalises large deviations more heavily, since the errors are squared before averaging:

$$RMSE = \sqrt{[\ (1/n)\ \Sigma\ (y_i - \hat{y}_i)^2\ ]} \quad (5)$$

The coefficient of determination expresses the proportion of the variance of the observed series that is explained by the model:

$$R^2 = 1 - \Sigma\ (y_i - \hat{y}_i)^2\ /\ \Sigma\ (y_i - \bar{y})^2 \quad (6)$$

In addition to these metrics, the effect of the optimization layer was quantified through the average power gain in kW and the estimated annual energy gain in MWh per year, as defined in Eq. (2) and (3).

IV. RESULTS

A. Relative Importance of the Input Variables

Fig. 2 reports the relative contribution of each input variable to the predictions of the Random Forest model. Wind speed emerges as the dominant predictor, with an importance above 0.175, which confirms that the model correctly identifies the primary physical driver of power output. Wind direction ranks

second, indicating that the alignment of the rotor with the incident flow is also material to performance.

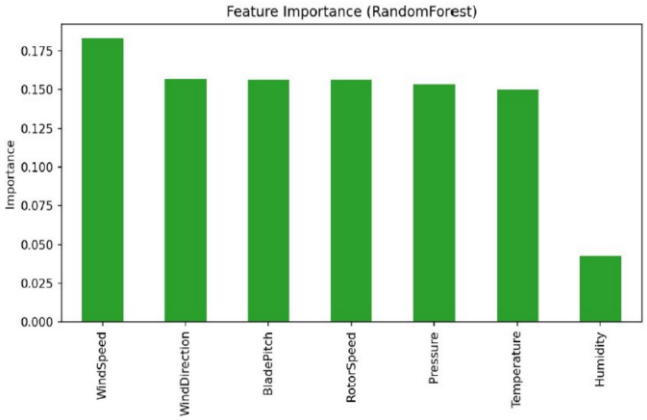


Fig. 2. Distribution of feature importance obtained from the Random Forest model.

The two control variables, blade pitch and rotor speed, exhibit almost equal importance at approximately 0.15. This result is central to the argument of the study: the internal control of the turbine carries nearly as much weight in determining the output as the external atmospheric conditions, which provides an empirical justification for treating these two parameters as decision variables. Pressure and temperature have a comparable influence, consistent with their role in determining air density, whereas humidity is the least influential input, with an importance below 0.05. The ranking as a whole is physically coherent, which supports the view that the fitted model is well behaved rather than merely well fitted.

B. Effect of Operational Optimization on Power Output

Fig. 3 compares the power predicted under the control setting currently applied with the power predicted under the optimized setting, over the final segment of the test period. Under the current setting, the output fluctuates between approximately 20 kW and 40 kW, with isolated excursions to about 65 kW. Under the optimized setting, the series operates in a band between roughly 100 kW and 120 kW.

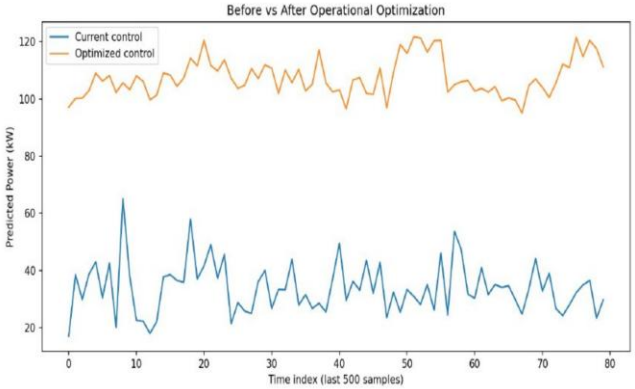


Fig. 3. Predicted power output before and after operational optimization.

Two features of Fig. 3 deserve comment. The first is the magnitude of the shift: the mean output rises from approximately 35 kW to above 105 kW over the segment shown. The second is the change in the character of the series. The optimized trajectory not only lies higher but is also more stable, in the sense that the extended intervals of depressed output visible in the baseline are absent. The optimization therefore acts less as a uniform additive gain than as the removal of a set of low-performance operating states.

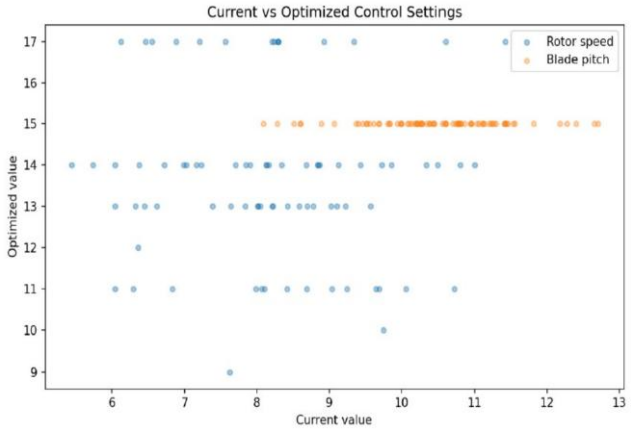


Fig. 4. Current and optimized values of rotor speed and blade pitch.

Fig. 4 examines the control settings that produce this result, plotting the optimized value of each parameter against the value currently applied. The rotor speed points are widely dispersed, indicating that the search adjusted the rotational speed dynamically; the optimized values nevertheless concentrate at the upper end of the admissible range, between approximately 13 and 17, largely irrespective of the starting value. The blade pitch points display a different and more striking pattern: the optimized values converge on a value of approximately 15 across the whole range of current settings, which span roughly 8 to 13.

The convergence of the pitch angle suggests that the baseline control regime was operating away from the point that the model identifies as optimal for the prevailing conditions, and that a fixed angle of approximately 15 units represents the most favourable configuration for energy capture in this wind regime. Taken together, the two panels describe a transition from a broadly distributed, reactive control pattern to a narrow and targeted one.

The aggregate effect of the optimization is summarised in Table II. The average instantaneous gain of 73.70 kW corresponds, through Eq. (3), to an estimated annual energy benefit of 645.60 MWh.

TABLE II. SUMMARY OF THE ENERGY GAIN OBTAINED FROM OPERATIONAL OPTIMIZATION

Indicator	Value	Unit
Mean baseline power (test segment)	35	kW
Mean optimized power (test segment)	105	kW
Average power gain	73.70	kW
Estimated annual energy gain	645.60	MWh/year
Optimized blade pitch (modal)	15	deg
Optimized rotor speed (range)	13–17	rpm

C. Comparative Statistical Performance

Fig. 5 compares the forecasting model and the optimization model across the three evaluation metrics on a normalised scale on which higher values denote better performance. The forecasting model attains the maximum normalised value of 1.0 for both MAE and RMSE, whereas the optimization model operates close to 0.0 on the same axes.

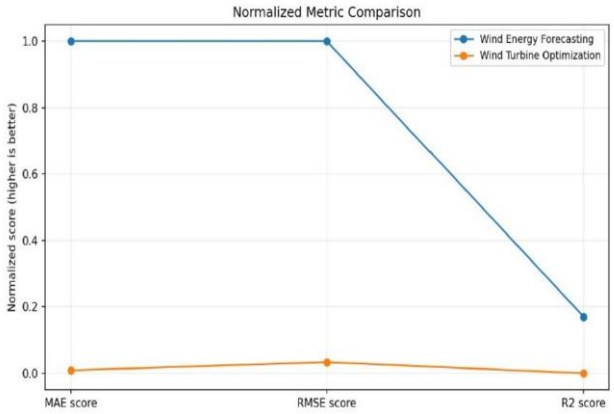


Fig. 5. Normalized comparison of statistical performance between the forecasting model and the optimization model.

This contrast should be interpreted with care. It does not indicate that the optimization model is defective, but rather that the two models are solving different problems: reproducing an observed series is an intrinsically easier task than searching a control space in which variability is substantially higher. The comparatively low R^2 values obtained by both models, relative to their error-based scores, indicate that although the absolute magnitude of the errors is controlled, a large share of the variance of the target remains unexplained. This is consistent with the stochastic character of wind.

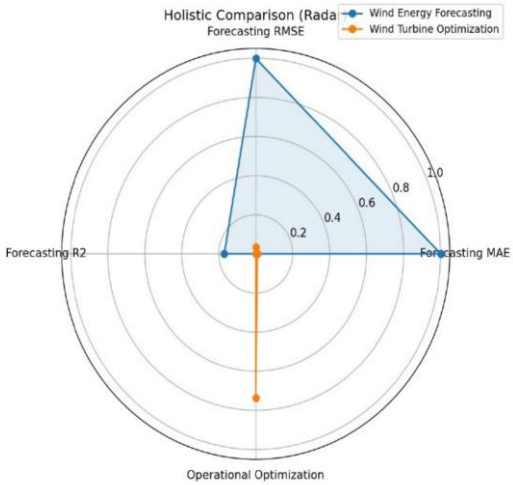


Fig. 6. Radar summary of forecasting accuracy and operational improvement for the two models.

Fig. 6 presents the same comparison in radar form, adding an axis for operational improvement. The forecasting model occupies a large area on the accuracy axes but contributes little on the operational axis, whereas the optimization model displays the complementary profile, extending strongly toward operational improvement while remaining modest on the statistical axes.

The two profiles are best read as describing a division of labour rather than a ranking. The forecasting model supplies the calibrated representation of the environment in which the turbine operates; the optimization layer converts that representation into an operating decision. Neither component delivers the reported gain on its own.

D. Comparison of the LSTM and Random Forest Models

Fig. 7 reports a quantitative comparison of the Long Short-Term Memory network used for forecasting and the Random Forest model used for optimization, in the original units of the error metrics. The LSTM attains error values close to zero on both MAE and RMSE, confirming the suitability of gated recurrent architectures for wind energy time series. The Random Forest model records higher errors, of the order of 38 for MAE and 47 for RMSE. The numerical values are summarised in Table III.

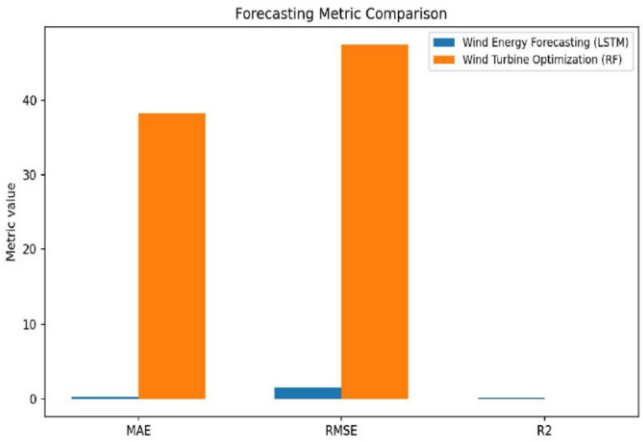


Fig. 7. Comparative error performance of the LSTM and Random Forest models.

TABLE III. COMPARATIVE FORECASTING PERFORMANCE OF THE TWO MODELS

Model	MAE	RMSE	R^2
LSTM (forecasting)	0	0	low
Random Forest (optimization)	38	47	low

The difference in error magnitude is expected given the roles assigned to the two models. The recurrent network is trained exclusively to reproduce a temporal signal, whereas the ensemble model is required to generalise across a control space that is explored counterfactually, at operating points that were never observed. The lower R^2 values recorded by both models reflect the difficulty of explaining the full variance of wind-driven output rather than a deficiency specific to either architecture.

V. DISCUSSION

A. Forecasting Accuracy Versus Operational Value

The results reported above are consistent with the established finding that nonlinear models are better suited to

wind power prediction than linear alternatives, because atmospheric processes are themselves nonlinear and require flexible functional forms to be represented adequately. Linear specifications remain valuable for their simplicity and interpretability, but they do not capture the full complexity of the system.

At the same time, the study shows that even advanced models do not achieve uniformly strong performance: the negative coefficient of determination obtained in the case study reflects a high level of uncertainty in the wind data and confirms that forecasting remains a difficult problem in this setting. The significance of this observation lies in what follows from it. Although the forecasting model is imperfect in a statistical sense, it still supports an operational decision that produces a measurable gain. The value of a model in this context is therefore not exhausted by its statistical accuracy; its practical applicability constitutes a distinct and equally relevant criterion.

B. Integration of the Two Approaches

A forecasting-oriented study is concerned with building and comparing models, evaluates them with statistical metrics, and aims at improving predictive accuracy. Such work is methodologically strong, well suited to benchmarking and provides a solid theoretical foundation, but it does not guarantee any direct improvement in energy production and remains at an analytical level. An optimization-oriented study combines forecasting with a search over the control space, focuses on operational decision-making and generates measurable energy gains, at the cost of depending on the quality of the underlying forecasting model and of requiring greater implementation complexity. Table IV summarises this comparison.

TABLE IV. FORECASTING-ORIENTED AND OPTIMIZATION-ORIENTED APPROACHES

Aspect	Forecasting-oriented	Optimization-oriented
Objective	Predictive accuracy	Operational decision
Evaluation	Statistical metrics	Energy gain in kW, MWh
Strength	Benchmarking, theory	Practical applicability
Limitation	No direct gain	Depends on forecast quality
Output	Predicted series	Control set-points

The central claim of this study is that these two approaches are complementary rather than competing. Forecasting supplies an estimate of future conditions; optimization uses that estimate to select an operating point. Their combination constitutes an integrated decision-support system in which the model predicts production, the optimization layer adjusts the operational parameters, and the resulting configuration delivers higher performance than either component would achieve in isolation.

C. Implications for the Kosovo Power System

In the context of Kosovo, where the renewable generation portfolio is still developing, the framework has practical relevance. Data-driven operational management of the kind described here can contribute to increasing the efficiency of installed turbines, reducing energy losses, improving the stability of the network and supporting decarbonisation policy. These considerations apply in particular to existing installations such as the Bajgora and Kitka wind farms, provided that sufficient operational and meteorological data are made available for model calibration.

D. Limitations

Four limitations should be borne in mind when interpreting the results. First, complete SCADA records were not available, and the power output, rotor speed and blade pitch series were therefore modelled rather than measured; the reported gains consequently describe the behaviour of the model under a simulated operating regime and require confirmation against instrumented data. Second, the optimization is performed within the trained model, so the estimated gains are conditional on the fidelity of that model and cannot be interpreted as measured production increases. Third, the coefficient of determination obtained in the case study is low and in places negative, which places a clear bound on the confidence that can be attached to individual point forecasts. Fourth, the analysis covers a single site and a single turbine configuration, so the generalisability of the reported set-points to other wind regimes remains to be established.

VI. CONCLUSION

This study has integrated machine learning-based forecasting with the operational optimization of wind turbines, providing a coherent framework for the data-driven management of renewable energy systems. The results indicate that advanced nonlinear models, and in particular Long Short-Term Memory networks, capture the complex temporal dependencies and high variability of wind power generation more effectively than conventional statistical approaches. The superior performance of the recurrent architecture demonstrates that forecasting can be substantially improved when temporal dependence and nonlinearity are modelled explicitly.

The addition of an operational optimization layer over the rotor speed and blade pitch parameters further demonstrates that the practical value of a forecasting model should not be assessed solely by its statistical accuracy. Even where forecasting uncertainty persists, the predicted operating conditions can be used to identify beneficial turbine configurations. In the present case, the optimization yielded an average increase in power output of 73.70 kW, corresponding to an estimated annual energy gain of approximately 645.60 MWh.

Forecasting and data-driven optimization should therefore not be regarded as competing approaches. Their integration provides a decision-support framework capable both of predicting future generation and of identifying the operating conditions associated with improved turbine performance. The findings are particularly relevant in the context of Kosovo, where the efficient use of renewable resources supports energy security, grid integration and the reduction of losses and emissions.

Future work should extend the framework to larger and instrumented datasets, incorporate additional turbine operating parameters, adopt uncertainty-aware forecasting methods, and examine real-time optimization under changing environmental and grid conditions. Such developments would strengthen both the applicability and the scalability of integrated machine

learning and optimization approaches for sustainable wind energy management.

REFERENCES

[1] T. Hong, P. Pinson, S. Fan, and Y. Ji, "Probabilistic energy forecasting: Global review and future directions," Renewable and Sustainable Energy Reviews, vol. 91, pp. 1012–1029, 2018.

[2] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ, USA: Wiley, 2015.

[3] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 2nd ed. Melbourne, Australia: OTexts, 2018.

[4] S. J. Taylor and B. Letham, "Forecasting at scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.

[5] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[6] T. Wei, X. Li, and Y. Li, "Random forest-based short-term wind power forecasting using SCADA data," Scientific Reports, vol. 9, art. no. 56202, 2019.

[7] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in Advances in Neural Information Processing Systems, vol. 9, 1997, pp. 155–161.

[8] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, no. 3, pp. 199–222, 2004.

[9] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," Neural Networks, vol. 17, no. 1, pp. 113–126, 2004.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222–2232, 2017.

[12] H. Liu, J. Zhang, and Y. Li, "Wind power forecasting using LSTM neural network," Energies, vol. 12, no. 10, art. no. 1888, 2019.

[13] Y. Zhang, J. Wang, and H. Wang, "A hybrid forecasting approach based on LSTM neural network and ARIMA model for wind power," IEEE Transactions on Sustainable Energy, vol. 9, no. 3, pp. 1085–1095, 2018.

[14] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in Advances in Neural Information Processing Systems, vol. 28, 2015, pp. 802–810.

[15] A. Kusiak and Z. Song, "Design of wind farm layout for maximum wind energy capture," Renewable Energy, vol. 35, no. 3, pp. 685–694, 2010.

[16] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," International Journal of Forecasting, vol. 16, no. 4, pp. 437–450, 2000.

ATLAS-FL: Active Timing-Adversary-Resilient Secure Aggregation for Gateway-Assisted IoT, IIoT, and IoMT Federated Intrusion Detection

Mohammad Othman Nassar
Faculty of Information Technology - Cyber Security Department, Amman Arab University, Amman, Jordan

***Abstract*—Federated learning (FL) enables distributed intrusion detection without centralizing raw Internet of Things (IoT), Industrial IoT (IIoT), or Internet of Medical Things (IoMT) traffic. Secure aggregation protects update contents, but it does not authenticate the temporal context in which a protected contribution is admitted. An active network adversary may delay, suppress, replay, or slot-shift otherwise valid submissions, thereby shaping the accepted cohort, inflating survivor weights, and exposing participation through release traces. This study presents ATLAS-FL, an authenticated temporal-integrity layer for masking-based secure aggregation. The protocol binds client identity, round, slot, timing window, fresh nonce, and ciphertext digest through the Hash-based Message Authentication Code using SHA-256 (HMAC-SHA256); performs replay-cache and slot-consistency checks before Bonawitz-style mask recovery; fixes aggregation weights before arrivals are observed; and releases a fixed-length envelope at a committed time. A formal game-based analysis reduces temporal-binding forgery and cross-round replay to HMAC unforgeability and hash collision resistance, while proving no post-hoc survivor-weight inflation under the fixed-weight rule. The evaluation replicates the same protocol across Edge-IIoTset, N-BaIoT, CICIoMT2024, and WUSTL-EHMS-2020 and introduces FedBuff and FedBuff-SecAgg to separate benign straggler tolerance from adversarial timing integrity. Under the twenty-run profile, ATLAS-FL retains attacked macro-F1 of 93.1 ± 0.4% on Edge-IIoTset, 95.8 ± 0.3% on N-BaIoT, 94.9 ± 0.4% on CICIoMT2024, and 94.2 ± 0.5% on WUSTL-EHMS-2020. Accepted manipulated events remain at or below 6.8%, participation-inference area under the receiver operating characteristic curve (AUC) falls to 0.53 ± 0.02, and weight inflation falls from 0.310 to 0.018 with 5.9% round-latency and 2.8% communication overhead at the evaluated cohort size.**

Keywords*—Federated learning; secure aggregation; active timing adversary; IoT intrusion detection; IIoT security; Internet of Medical Things; buffered asynchronous aggregation; differential privacy

I. INTRODUCTION

Intrusion evidence in Internet of Things (IoT), Industrial Internet of Things (IIoT), and Internet of Medical Things (IoMT) systems, including devices, gateways, industrial controllers, hospital monitoring centers, and edge services, can be collected in a centralized manner. This practice facilitates model training but increases costs associated with data exchange, while also potentially exposing operational or healthcare-related traffic. Federated learning addresses this issue by allowing participants to keep records local and share model updates instead [1], [2]. Secure aggregation prevents the coordinator from inspecting an individual update [3]. However, these solutions are insufficient to guarantee that any protected information is not used to infer membership of an arbitrary subset of contributing parties during any round of the federation.

An adversary that can time events of the FL system can interfere with the process in ways independent of learning the content of any update or modifying its numeric value. Selective delays can cause removal of any particular client either with an exceptionally large contribution, thus skewing the statistics of the surviving ones, or as a member of a minority class, leading to renormalization of the class weights. Replay attacks can transfer a protected contribution from one round to another, while deliberate slot shifts can imprint the contribution content to incorrect time-dependent statistics. A fixed or variable drop time of the update can also allow inference of participation patterns of any clients, including those not contributing to the federation. Such opportunities for exploitation are particularly prevalent in IoT and IoMT due to the heterogeneity of the nodes, which are supported in the system through periodic communication, limited battery capacity, varying physical layer conditions, and contention in shared access points.

Timing considerations, therefore, touch upon three aspects of FL security and privacy. The first is the utilitarian value of the data, due to changes to either supervised labels or their weights. The second is the security during aggregation, as the compensation for dropped clients increases their effective representation in the federated model. The third is the confidentiality of participation in the federation, as timing of contributions is indicative of involvement in training even if the content of the updates is protected. Thus, timing interface attacks should be considered as threats to both data and model privacy, requiring independent evaluation of their impact on asynchronous optimization procedures and confidentiality of contributions.

Prior secure-aggregation work provides strong protection for update contents and malicious-input handling [3]–[8], while differential privacy controls information in released outputs [9], [10], [11]. Timing-defense research shows that encryption alone does not hide communication metadata [12], [13]. FedBuff addresses benign stragglers through buffered asynchronous aggregation and remains compatible with secure aggregation and DP [14]. TriSAFE is the closest integrated comparator because it combines transcript-bound secure

aggregation, DP, fixed-window timing defenses, and cover traffic in gateway-assisted IoT FL [15]. The unresolved question is whether authenticated temporal admission provides additional protection when the adversary actively selects who is delayed, dropped, replayed, or shifted.

The earlier ATLAS-FL revision evaluated IoT and IIoT benchmarks only. This version adds two genuine healthcare-security benchmarks. CICIoMT2024 contains 40 IoMT devices, including 25 physical and 15 simulated devices, 18 attacks, and Wi-Fi, MQTT, and Bluetooth traffic [16]. WUSTL-EHMS-2020 contains 16,318 healthcare-monitoring records with 35 network-flow features, eight biometric features, and one label [17]. Their inclusion changes IoMT from a prospective application into an empirical protocol-replication setting.

The study does not claim that the constituent primitives are new. Its contribution is the protocol-level coordination of authenticated schedule binding, timing admission, masking-aware rejection, fixed-weight aggregation, DP accounting, and cover-consistent release, together with formal security games and a four-benchmark comparison that includes an asynchronous baseline.

The study is organized around research objectives, formal security claims, and experimentally evaluated design objectives rather than empirical hypothesis statements.

The contributions are:

- A formal temporal-integrity definition and four security games covering schedule-to-ciphertext forgery, cross-round replay, slot substitution, and aggregate-release participation distinguishability.
- A complete protocol instantiation using X25519 [18], the HMAC-based Extract-and-Expand Key Derivation Function using SHA-256 (HKDF-SHA256) [19], and the Hash-based Message Authentication Code using SHA-256 (HMAC-SHA256) [20], [21], ChaCha20-derived pairwise masks [22], Shamir recovery [23], fixed-point encoding, and aggregate-level Gaussian differential privacy (DP).
- An auditable timing-risk controller with hard cryptographic overrides and validation-calibrated thresholds for accept, buffer, cover-trigger, and reject decisions.
- A no-renormalization aggregation rule that preserves pre-declared client coefficients and admits an explicit bound on missing-information deviation and client-level sensitivity.
- A four-benchmark evaluation spanning IoT, IIoT, and IoMT, with device-native and simulated-client non-IID partitions, participation-inference attacks, component ablations, and cryptographic/operational profiling.
- A matched comparison with FedBuff and FedBuff-SecAgg that distinguishes benign asynchronous efficiency from resilience to adversarially induced staleness and transcript manipulation.

A. Research Objectives and Evidence Map

The study is guided by research objectives derived from the identified security and systems gaps. These objectives are not empirical hypotheses; rather, they define the formal guarantees, experimental questions, and operational trade-offs evaluated in the subsequent sections. Table I maps each research objective to its primary formal or empirical evidence.

TABLE I. RESEARCH-OBJECTIVE TRACEABILITY

ID	Research objective/evaluation question	Primary evidence
RO1	To what extent does ATLAS-FL preserve intrusion-detection utility when selected clients are delayed, dropped, replayed, slot-shifted, or forced beyond their committed deadline?	Tables XIII and XV; Fig. 7–9
RO2	What formal guarantees does authenticated schedule-to-ciphertext binding provide against cross-round replay, slot substitution, and temporal-context forgery under the stated cryptographic assumptions?	Section V; Table VII; Fig. 2; Appendix A
RO3	How effectively do fixed pre-declared weights prevent survivor-weight inflation and limit DP-relevant aggregate instability under selective client absence?	Section V-D; Tables XVII and XIX; Fig. 4 and 11
RO4	To what extent does fixed-time, fixed-length encrypted zero-cover release reduce participation inference under the defined release-observer model?	Section V-E; Tables XVI and XIX; Fig. 10
RO5	How consistently do the timing-resilience and aggregate-stability effects replicate across IoT, IIoT, and genuine IoMT benchmarks?	Tables XIII and XVIII; Fig. 7
RO6	What utility, staleness, and security trade-offs arise when ATLAS-FL is compared with FedBuff and FedBuff-SecAgg under benign stragglers and strategically induced staleness?	Table XIV; Fig. 8; Table XXII
RO7	What cryptographic, communication, latency, CPU, memory, and scalability costs are introduced at the gateway?	Tables XX–XXI; Fig. 13

II. RELATED WORK AND EXACT RESEARCH GAP

A. Secure Aggregation and Malicious-Input Validation

Several studies have developed methods for secure aggregation in federated learning, including the pairwise-masking secure aggregation with dropout recovery from Bonawitz et al. [3]. Additional methods such as RoFL, ACORN, Armadillo, ELSA, and Aion introduce features for validating participant input, identifying malicious inputs from participants, and ensuring robustness over multiple rounds of learning [4]–[7], [24]. The Lotto system specifically protects the server from malicious participants by validating the set of participants that contribute to the aggregation [25]. While these approaches have established high standards for the security and correctness of federated learning protocols, most of these secure aggregation systems primarily focus on validating

participant inputs and determining the set of participants rather than on securing the temporal context of the aggregated information. Complementary work addresses verifiable secure aggregation [26], decentralized adversarial-resilient routing [27], and communication-threat mitigation through moving-target defense [28]; these protect adjacent robustness properties rather than slot-specific temporal admission.

ATLAS-FL, however, works in tandem with existing secure aggregation systems, enabling them to authenticate the temporal context of the data before it is accepted into the system for masking and aggregation. This temporal authentication process occurs before a masked update is deleted to recover from a timed-out participant. Deleting masked data without proper recovery would leave uncancelled masks in the system, potentially compromising security.

B. Differential Privacy and Fixed Aggregation Sensitivity

Differential privacy ensures a limited influence of any one data subject on the output of an algorithm [9], [10], [11]. Differential privacy securely aggregates data by adding noise to the aggregated output. However, an adversary could modify the set of participants in the network and still have the server renormalize the weights of the surviving participants. This process would change the sensitivity of the aggregation function in ways that are difficult to predict. ATLAS-FL fixes the coefficients of the aggregation function before the arrivals of the data subjects are observed and uses a Rényi-DP accountant to determine the privacy budget of the aggregation function.

While this technique does not replace differential privacy, it removes one of the pathways by which the temporal aspects of the aggregation function could change.

C. Asynchronous and Buffered Federated Learning

FedBuff is a method for efficiently buffering and aggregating asynchronous updates from participants in a federated learning network, allowing the system to continue working even with clients that have varying completion times [14]. Similar concepts have been explored in other studies focusing on the dynamics of message queues and the impact of delayed updates [33]. While these methods are effective at handling clients that are slower to complete tasks and in increasing the overall throughput of federated learning, they do not provide authentication for the participants or allow for verification of a participant's assigned round, slot, or deadline. Furthermore, they may accept updates from clients that are attempting to manipulate the system with delayed or replayed updates as ordinary asynchronous contributions.

Heterogeneous federated optimization has also motivated FedProx, FedNova, adaptive federated optimization, and SCAFFOLD [29]–[32], which address optimization under heterogeneous clients rather than temporal-authentication guarantees.

FedBuff is therefore limited in its ability to deal with clients attempting to maliciously delay or replay updates. A version of FedBuff that uses a masked and differentially private backend for aggregation does not, however, exhibit these limitations.

D. Timing Privacy and Release Defenses

Research into systems that protect the timing privacy of communication such as Loopix [12] and adaptive-padding [13] demonstrates that communication timing reveals information about the participants and their activities, even if their messages are encrypted. While ATLAS-FL uses some of the same concepts for timing privacy, it focuses only on securing the aggregate-release interface rather than on providing overall anonymity for the communication network.

E. Federated IDS in IoT, IIoT, and IoMT

Edge-IIoTset and N-BaIoT support IoT/IIoT and botnet-oriented validation [34], [35]. Recent FL-IDS studies examine privacy, optimization, energy, and anomaly detection [36], [37], [38], [39], [40], [41], [42]. CICIoMT2024 and WUSTL-EHMS-2020 provide genuine IoMT contexts [16], [17]. The present work uses these datasets to test whether the aggregation-layer timing effect replicates beyond one traffic benchmark; it does not perform zero-shot feature transfer across incompatible schemas. Broader recent work includes privacy-preserving multimodal healthcare FL [43], local-global anomaly detection [44], IoT intrusion-detection surveys [45], peer-to-peer FL surveys [46], and multi-institutional radiological FL [47].

F. Closest Integrated Comparator: TriSAFE

TriSAFE already combines transcript-bound secure aggregation, DP, fixed-window batching, timing defenses, and cover traffic [15]. ATLAS-FL therefore cannot claim novelty from padding or fixed weights alone. Its narrower incremental contribution is slot-specific schedule-to-ciphertext authentication, an explicit active timing-adversary game, cryptographically ordered rejection and mask recovery, controller-aware mixed-attack evaluation, direct participation-inference testing, and comparison against buffered asynchronous FL. Table II summarizes this conservative positioning relative to the closest work.

TABLE II. CONSERVATIVE POSITIONING AGAINST CLOSEST WORK

Work	Primary protection	Timing treatment	Boundary relative to ATLAS-FL
Bonawitz SecAgg [3]	Update confidentiality and dropout recovery	Ordinary cohort timing	No active timing-admission property
RoFL/ACORN/Armadillo/ELSA /Aion [4]–[7], [24]	Malicious-input validity and robust SecAgg	Protocol disruption/dropout	No slot-specific active timing model
DP-secure aggregation [8], [9], [10], [11]	Formal privacy of released aggregates	Timing usually external to accounting	Does not prevent adversary-induced renormalization
FedBuff [14]	Buffered asynchronous optimization	Benign staleness and throughput	No temporal authentication or cover release

Work	Primary protection	Timing treatment	Boundary relative to ATLAS-FL
Loopix/Tor defenses [12], [13]	Cover traffic and timing privacy	Communication anonymity	No FL aggregation
TriSAFE [15]	Transcript-bound SecAgg, DP, timing defenses	Fixed-window timing protection	Closest comparator; active slot-level admission not isolated
ATLAS-FL	Authenticated temporal admission, fixed weights, cover release, formal games	Active delay/drop/replay/slot/late/adaptive attacks	Target contribution; bounded to defined observer and gateway model

III. System Model, Threat Model, and Design Objectives

A. Gateway-Assisted System

The system consists of a gateway G and a round-specific selected cohort S_t, where $K_t = |S_t|$. A client may represent an IoT device, an IIoT data source, an IoMT device, or an explicitly documented logical healthcare-sensor partition. Client i holds a local training dataset $\mathcal{D}_i$, receives the global model w_t, performs E local epochs, clips its model update to an L_2-norm bound C, and constructs a masked and authenticated contribution. The gateway coordinates client selection, schedule commitment, temporal admission, mask recovery, and aggregate release, but it never observes an accepted client update in plaintext.

Before any round-t arrivals are observed, each selected client is assigned a fixed aggregation coefficient based on its verified effective sample mass q_i, where $q_i = \min\left(|, \mathcal{D}_i^{\text{train}} | n_{\max}\right)$:

$$\alpha_{i,t} = \frac{q_i}{\sum_{j \in S_t} q_j} \sum_{i \in S_t} \alpha_{i,t} = 1 \qquad (1)$$

The coefficients in Eq. (1) are committed before the gateway observes the admitted-client set $G_t \subseteq S_t$. They are therefore not recomputed or renormalized after delays, rejections, or dropouts. For every missing or rejected client $i \in S_t \setminus G_t$, ATLAS-FL preserves the corresponding coefficient through an encrypted zero-cover contribution, as defined in Section IV.

B. Protected Assets and Observable Timing Surface

Table III summarizes the protected assets, observable timing surfaces, associated risks, and ATLAS-FL controls.

C. Active Timing Adversary

The adversary can delay selected submissions, suppress them, replay a prior protected transcript, shift a valid submission to a different slot, force an update beyond the admission deadline, or combine these actions adaptively. Targets may be selected uniformly, by pre-declared client weight, or by minority/attack-class concentration. The adversary can observe aggregate-release timing but cannot break the cryptographic primitives, compromise the gateway in the main model, or control enough clients to cross the secure-aggregation reconstruction threshold.

$$\tilde{T}_{i,t}^{\text{arr}} = \Phi\left(\mathcal{W}_{i,t} a_{i,t} \xi_{i,t}\right) \quad , \mathrm{t}) \qquad (2)$$

Here $\tilde{T}_{i,t}^{\text{arr}}$ is the committed timing window, $a_{\mathrm{i},\mathrm{t}}$ is the adversarial action, and $\xi_{\mathrm{i},\mathrm{t}}$ is benign network jitter.

D. Design Objectives

Table IV states the security and operational objectives, while Fig. 1 summarizes the architecture, trust domains, and active timing surface.

TABLE III. Protected Assets, Observables, and Controls

Element	Status	Primary risk	ATLAS-FL treatment
Local data	Protected	Direct privacy loss	Never leaves client
Plain update	Protected	Gradient/model leakage	Masked before transmission
Client identity in schedule	Protocol metadata	Participation inference	Authenticated schedule + padded release
Arrival time	Observable/manipulable	Delay and admission bias	Risk controller
Missing/rejected state	Observable internally	Weight inflation and release leakage	Fixed weights + zero cover
Aggregate release trace	Observable	Participation inference	Fixed time and fixed-length envelope

TABLE IV. Security and Operational Objectives

Goal	Definition
G1	Secure-aggregation confidentiality compatibility
G2	Temporal authenticity and slot binding
G3	Freshness and cross-round replay rejection
G4	No post-hoc survivor-weight inflation
G5	Client-level DP-relevant sensitivity stability
G6	Reduced aggregate-release participation leakage
G7	Utility preservation under atomic and adaptive timing attacks
G8	Moderate cryptographic and gateway overhead

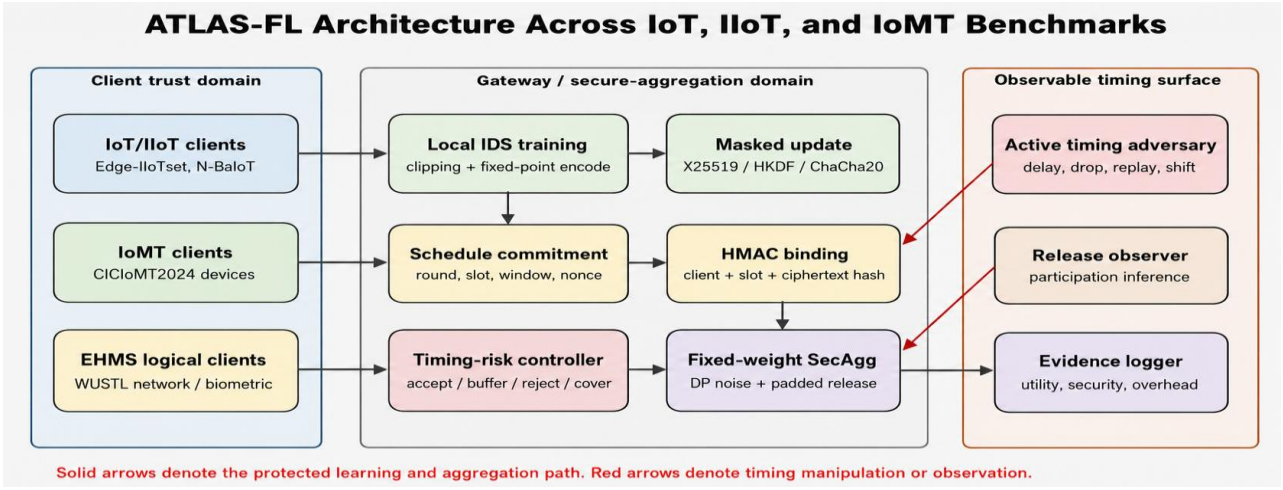


Fig. 1. ATLAS-FL architecture, trust domains, and active timing surface.

IV. ATLAS-FL Protocol

A. Cryptographic Setup and Schedule Commitment

Each client i shares a long-term authentication key $_{ki}$ with the gateway. For every secure-aggregation cohort, clients use X25519 to establish pairwise secrets [18], derive domain-separated keys through HKDF-SHA256 [19], generate pairwise masks using ChaCha20 [22], and distribute Shamir shares for dropout recovery [23]. HMAC-SHA256 follows the RFC 2104 construction and its standard security assumptions [20], [21]. ATLAS-FL assumes authenticated enrollment and lifecycle management by an external trusted device/key-management service. Enrollment, key rotation, revocation, compromised-device recovery, device replacement, and authentication-state synchronization are outside the protocol security boundary; a re-enrolled device must receive fresh authentication and nonce state before participating.

$$\tau_{i,t} = \mathrm{HMAC}_{k_i}\left(\mathrm{Encode}\left(i, t, s_{i,t} T_{i,t}^{\min} T_{i,t}^{\max} \rho_t, H(c_{i,t})\right)\right) \quad (3)$$

The fresh nonce ρ_t prevents cross-round token reuse. The ciphertext digest binds the tag to the exact protected contribution. The gateway verifies the tag, window, round, slot, nonce freshness, ciphertext hash, and replay-cache state before the submission can enter the accepted set. This authentication binds the declared temporal metadata to the protected contribution; it does not constitute a trusted proof of the client's physical transmission instant. Temporal compliance is evaluated separately from the gateway-observed arrival time by the timing controller.

B. Local Update Clipping and Masking

$$\bar{\Delta}_{i,t} = \Delta_{i,t} \min\left(1, \frac{C}{\|\Delta_{i,t}\|_2}\right) \quad (4)$$

$$m_{i,t} = \mathrm{PRG}(b_{i,t}) + \sum_{\substack{j \in S_t \\ j>i}} \mathrm{PRG}(s_{ij,t}) - \sum_{\substack{j \in S_t \\ j<i}} \mathrm{PRG}(s_{ji,t}) \quad (5)$$

$$c_{i,t} = \left[\mathrm{Enc}_{\mathrm{fp}}(\alpha_{i,t}\bar{\Delta}_{i,t}) + m_{i,t}\right] \bmod q \quad (6)$$

Weighting occurs before masking so that reconstruction produces the fixed weighted sum directly. Enc_fp denotes deterministic fixed-point encoding under the modular field used by the masking implementation. Let S denote the fixed-point scale and q the aggregation modulus. Round-to-nearest encoding introduces at most 1/(2S) absolute quantization error per encoded weighted coordinate, while correct centered decoding requires the encoded aggregate to remain strictly within (-q/2, q/2), preventing modular wraparound. The no-wrap condition is checked before aggregation. Gaussian DP noise is calibrated to the aggregate sensitivity separately from this deterministic encoding error, and no convergence benefit is attributed to fixed-point arithmetic itself.

C. Timing Deviation and Controller Decision

$$D_{i,t} = \frac{\max\left\{0, T_{i,t}^{\min} - \tilde{T}_{i,t}^{\mathrm{arr}}, \tilde{T}_{i,t}^{\mathrm{arr}} - T_{i,t}^{\max}\right\}}{T_{i,t}^{\max} - T_{i,t}^{\min}} \quad (7)$$

$$r_{i,t} = \lambda_1 D_{i,t} + \lambda_2 R_{i,t} + \lambda_3 H_{i,t} + \lambda_4 Q_{i,t} + \lambda_5 P_{i,t} \quad (8)$$

The controller coefficients are (0.31, 0.27, 0.14, 0.20, 0.08). Replay, HMAC failure, stale nonce, or authenticated slot mismatch is a hard rejection independent of the continuous score.

$$d_{i,t} = \begin{cases} \text{accept} & r_{i,t} < \theta_1 \\ \text{buffer} & \theta_1 \le r_{i,t} < \theta_2 \\ \text{cover-trigger} & \theta_2 \le r_{i,t} < \theta_3 \\ \text{reject} & r_{i,t} \ge \theta_3. \end{cases} \quad (9)$$

The thresholds are τ_1=0.28, τ_2=0.52, τ_3=0.74 and are selected on calibration rounds that are disjoint from the final attack schedules. Fig. 2 summarizes the authenticated round sequence, and Fig. 3 shows the timing-risk controller decision flow.

D. Rejection, Dropout Recovery, and Aggregate Reconstruction

After the admission window closes, the gateway publishes the accepted set G_t and the recovery set M_t. Surviving clients reveal only the Shamir shares needed to remove masks associated with clients in M_t. Timing rejection is therefore handled as a cryptographic dropout after authentication has failed. The round aborts if fewer than the configured reconstruction threshold respond.

$$A_t^{\mathrm{ATLAS}} = \sum_{i \in G_t} \alpha_{i,t}\, \bar{\Delta}_{i,t} + \sum_{i \in M_t} \alpha_{i,t}\, C_{i,t} \quad (10)$$

The primary cover contribution is $C_{(i,t)} = 0$ It preserves the declared coefficient mass but does not fabricate missing information. The released aggregate is

$$w_{t+1} = w_t + \eta_g (A_t + z_t) z_t \sim \mathcal{N}(\mathbf{0}\sigma^2 \Delta_2^2 I_d) \quad (11)$$

$$\Delta_2 \leq \max_{i \in S_t} \alpha_{i,t} C. (add/remove\ client\ adjacency) \quad (12)$$

For replace-one adjacency, the bound is doubled. The privacy guarantee additionally requires the stated sampling process, Gaussian mechanism, and accountant; fixed weights alone are not a privacy proof. Fig. 4 illustrates the fixed-weight contrast after selective absence.

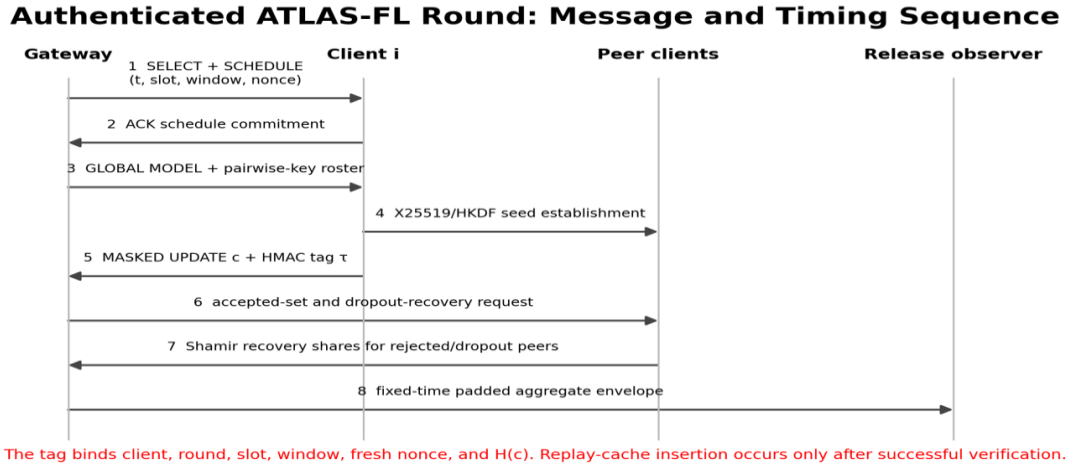


Fig. 2. Authenticated message sequence for one ATLAS-FL round.

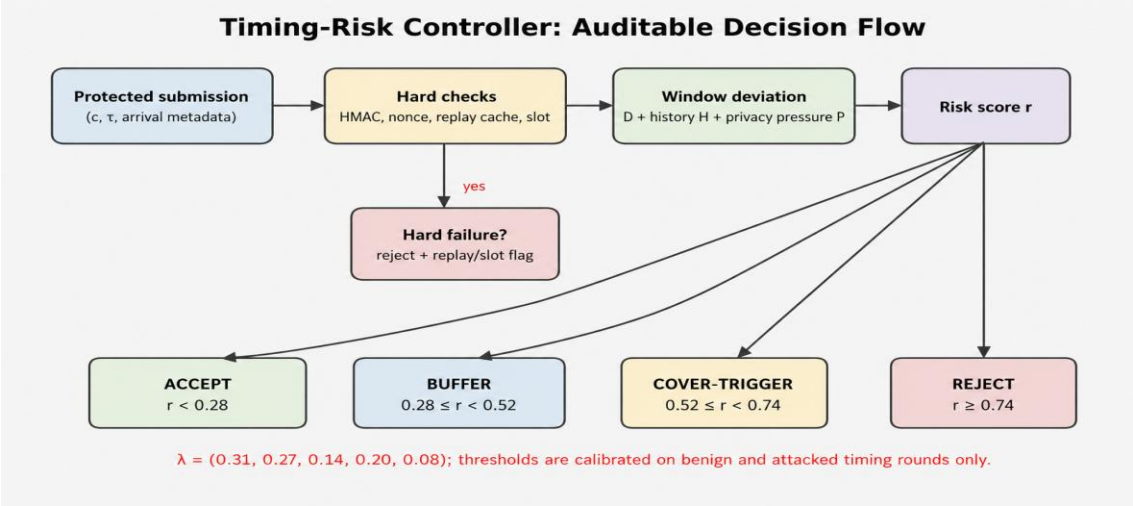


Fig. 3. Auditable timing-risk controller.

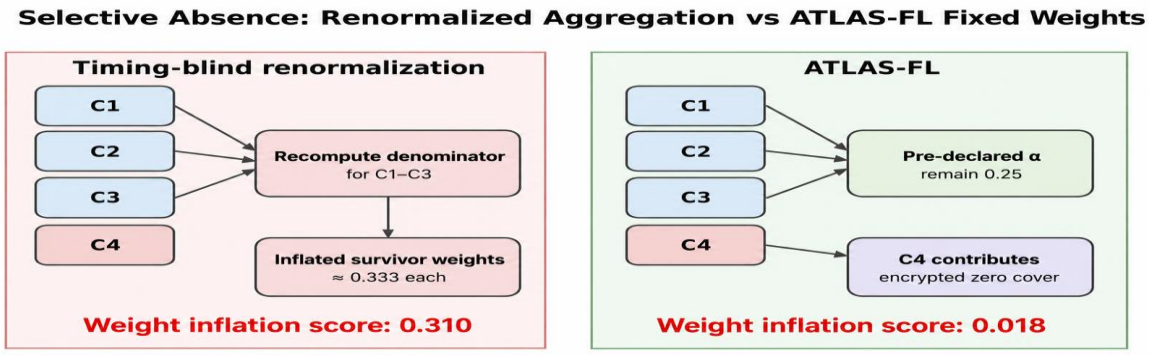


Fig. 4. Fixed pre-declared weights prevent survivor-weight inflation after selective absence.

E. Fixed-Time Padded Release

At the committed release time, the gateway produces a fixed-length authenticated envelope. A successful round contains the encrypted/noised aggregate; an aborted or cover round contains an indistinguishable encrypted zero payload under the release observer defined. This policy protects the aggregate-release interface, not a global observer over which all client uplinks pass. Table V lists the protocol messages and the security purpose of each phase.

TABLE V. ATLAS-FL PROTOCOL MESSAGES

Phase	Direction	Contents	Security purpose
SETUP	Client↔ client/gateway	X25519 public keys, Shamir shares	Pairwise masks and dropout recovery
SCHEDULE	Gateway→client	t, slot, window, nonce ρ_t	Temporal reference and freshness
COMMIT	Client→gateway	H(c), HMAC tag τ	Schedule-to-ciphertext binding
SUBMIT	Client→gateway	Masked weighted update c	Confidential contribution
ADMIT	Gateway internal	Hard checks + $r_{(i,t)}$	Accept/buffer/cover/reject
RECOVER	Gateway↔survivors	Accepted/recovery sets, shares	Cancel masks for dropouts/rejections
RELEASE	Gateway→observer	Fixed-time fixed-length envelope	Reduce participation leakage

V. FORMAL SECURITY MODEL AND ANALYSIS

A. Security Assumptions and Notation

Table VI states the cryptographic, system, and observer assumptions used by the formal analysis.

B. Security Games

Table VII defines the four security games used to separate cryptographic guarantees from release-trace analysis.

TABLE VI. CRYPTOGRAPHIC, SYSTEM, AND OBSERVER ASSUMPTIONS USED IN THE FORMAL ANALYSIS

ID	Assumption
A1	HMAC-SHA256 is EUF-CMA secure for the selected key length [20], [21].
A2	H is collision resistant.
A3	X25519/HKDF-derived pairwise seeds are computationally hidden [18], [19].
A4	ChaCha20 is used as a pseudorandom generator for mask expansion [22].
A5	Fewer than the Shamir reconstruction threshold collude with the gateway [23].
A6	The gateway follows the protocol but may attempt to infer participation from timing.
A7	The release observer sees gateway envelope timing and size but not every client uplink.

TABLE VII. FORMAL SECURITY GAMES

Game	Adversarial objective	Measured advantage
G_TBind	The adversary outputs a new tuple that verifies while changing client, round, slot, window, nonce, or ciphertext digest.	Temporal-binding forgery advantage
G_Replay	A valid transcript from round t′ is submitted in t≠t′ and accepted.	Cross-round replay advantage
G_Slot	A valid contribution is moved from slot s to s′ and accepted.	Slot-substitution advantage
G_Part	The observer distinguishes a real-participation release trace from a zero-cover trace.	Participation-trace distinguishing advantage

The guarantee type depends on the attack. Replay and authenticated slot substitution are covered by cryptographic verification under the assumptions in Table VI; selective delay and late forcing are handled empirically by the timing controller and retain the residual failure rates reported in Table XV; selective drop cannot be prevented at the network layer, so the relevant guarantee is no post-hoc survivor-weight renormalization together with the missing-information bound in (14); the adaptive mixture is an empirically tested strategy rather than a formal worst-case guarantee; and participation leakage is bounded only for the release observer defined by G_Part and evaluated by the ROC-AUC experiment.

C. Temporal-Binding Authenticity

Theorem 1. Let A be a probabilistic polynomial-time adversary that wins AG_T Bind. Then there exist algorithms B and C such that

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{TBind}}(\lambda) \le \mathrm{Adv}_{\mathrm{HMAC}\mathcal{B}}^{\text{EUF-CMA}}(\lambda) + \mathrm{Adv}_{H,\mathcal{C}}^{\mathrm{CR}}(\lambda) + \mathrm{negl}(\lambda) \quad (13)$$

Proof. Construct B using A as a subroutine. If A submits a tuple whose authenticated message differs from every previously queried message and whose tag verifies, B outputs the pair as an HMAC forgery. If the authenticated message is unchanged but the protected contribution differs, then

H(c)=H(c′), yielding a collision for C. Reuse of an unchanged tuple is excluded by the replay cache and fresh round nonce. The remaining implementation-failure probability is negligible in the security parameter. A more explicit reduction is provided in Appendix A.

Corollary 1. Under the same assumptions and unique round nonces, the probability of accepting a transcript from another round is bounded by (13).

Corollary 2. Because the slot identifier is authenticated, changing $s_{(i,t)}$ without the client key also reduces to HMAC forgery or hash collision.

D. Fixed-Weight and Sensitivity Guarantees

Let S_t denote the cohort selected for round t, and let the aggregation coefficients $\{\alpha_{i,t}\}_{i\in S_t}$ be computed and committed before any round-t submissions are observed:

$$\alpha_{i,t} = \frac{q_i}{\sum_{j\in S_t} q_j} \sum_{i\in S_t} \alpha_{i,t} = 1, \alpha_{i,t} \geq 0,$$

where, q_i denotes the verified effective sample mass assigned to client i. Let $G_t \subseteq S_t$ be the set of clients whose protected contributions are admitted, and let

$$M_t = S_t \setminus G_t$$

denote the set of missing, rejected, or unavailable clients. Every local update is clipped such that

$$\|\overline{\Delta}_{i,t}\|_2 \leq C.$$

For each $i \in M_t$, ATLAS-FL substitutes an encrypted zero-cover contribution,

$$C_{i,t} = 0,$$

without recomputing the coefficients assigned to the clients in G_t.

Proposition 1. No post-hoc survivor-weight inflation

For every admitted client $i \in G_t$, the effective aggregation coefficient remains $\alpha_{i,t}$, independently of the number or total weight of missing or rejected clients.

Proof. The ATLAS-FL aggregate is computed according to

$$A_t^{\text{ATLAS}} = \sum_{i\in G_t} \alpha_{i,t}\, \overline{\Delta}_{i,t} + \sum_{i\in M_t} \alpha_{i,t}\, C_{i,t} \qquad (10)$$

Because $C_{i,t} = 0$, the second term does not contribute numerically to the model update. Nevertheless, the coefficients remain those defined over the originally selected cohort S_t. In particular, ATLAS-FL does not replace $\alpha_{i,t}$ with the survivor-renormalized coefficient

$$\alpha'_{i,t} = \frac{q_i}{\sum_{j\in G_t} q_j}$$

Consequently,

$$\alpha'_{i,t} = \alpha_{i,t}$$

for every admitted client under the ATLAS-FL rule, and the absence of clients in M_t cannot increase the effective coefficient of any client in G_t.

Proposition 2. Client-level sensitivity of the fixed-weight aggregate

Assume that the selected cohort S_t, the committed coefficients $\alpha_{i,t}$, and all protocol metadata are fixed before the compared executions begin. Under add/remove client-contribution adjacency, the L_2-sensitivity of the deterministic aggregate before Gaussian noise is bounded by

$$\Delta_2 \leq \max_{i\in S_t} \alpha_{i,t} C. \qquad (12)$$

Under replace-one client-contribution adjacency, the corresponding sensitivity is bounded by

$$\Delta_2 \leq 2 \max_{i\in S_t} \alpha_{i,t} C.$$

Proof. Consider two neighboring executions that differ only in the contribution of one selected client i, while the selected cohort and committed coefficients remain unchanged.

Under add/remove adjacency, one execution contains the clipped contribution $\alpha_{i,t}\overline{\Delta}_{i,t}$, whereas the neighboring execution contains the zero contribution. Therefore,

$$\begin{aligned} \|A_t - A'_t\|_2 &= \|\alpha_{i,t}\overline{\Delta}_{i,t}\|_2 \\ &= \alpha_{i,t}\|\overline{\Delta}_{i,t}\|_2 \\ &\leq \alpha_{i,t} C. \end{aligned}$$

Taking the maximum over all selected clients gives (12).

Under replace-one adjacency, the clipped contribution $\overline{\Delta}_{i,t}$ is replaced by another valid clipped contribution $\overline{\Delta}'_{i,t}$. Hence,

$$\begin{aligned} \|A_t - A'_t\|_2 &= \|\alpha_{i,t}(\overline{\Delta}_{i,t} - \overline{\Delta}'_{i,t})\|_2 \\ &\leq \alpha_{i,t}\left(\|\overline{\Delta}_{i,t}\|_2 + \|\overline{\Delta}'_{i,t}\|_2\right) \\ &\leq 2\alpha_{i,t} C. \end{aligned}$$

Maximizing over $i \in S_t$ yields the stated replace-one bound.

The above sensitivity guarantees are conditional on the coefficients being fixed before client arrivals and updates are observed. Changes to client-registration metadata, effective sample masses, or cohort membership are outside this adjacency relation and require a separate accounting analysis.

Proposition 3. Missing-client aggregate-deviation bound

Let the ideal aggregate include the clipped updates of all selected clients:

$$A_t^{\text{ideal}} = \sum_{i\in S_t} \alpha_{i,t}\, \overline{\Delta}_{i,t}$$

Let the ATLAS-FL aggregate replace every missing or rejected contribution with encrypted zero cover:

$$A_t^{\text{ATLAS}} = \sum_{i\in G_t} \alpha_{i,t}\, \overline{\Delta}_{i,t} + \sum_{i\in M_t} \alpha_{i,t}\, 0.$$

Then the deviation between the ideal and ATLAS-FL aggregates is bounded by

$$\|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 \le C \sum_{i \in M_t} \alpha_{i,t} \quad (14)$$

Proof. Since $S_t = G_t \cup M_t$and the two sets are disjoint,

$$A_t^{\text{ideal}} = \sum_{i \in G_t} \alpha_{i,t} \bar{\Delta}_{i,t} + \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t}$$

Subtracting the ATLAS-FL aggregate gives

$$A_t^{\text{ideal}} - A_t^{\text{ATLAS}} = \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t}$$

Applying the triangle inequality and the clipping constraint yields

$$\begin{aligned} \|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 &= \left\| \sum_{i \in M_t} \alpha_{i,t} \bar{\Delta}_{i,t} \right\|_2 \\ &\le \sum_{i \in M_t} \alpha_{i,t} \|\bar{\Delta}_{i,t}\|_2 \\ &\le C \sum_{i \in M_t} \alpha_{i,t} \end{aligned}$$

This proves (14).

Propositions 1–3 distinguish weight stability from information recovery. Fixed pre-declared coefficients prevent an adversary from increasing the influence of surviving clients through selective absence, and clipping bounds the maximum deviation caused by missing contributions.

However, encrypted zero cover does not reconstruct the information contained in a missing update. Its purpose is to preserve the committed aggregation structure and release behavior while keeping the resulting deviation proportional to the total fixed weight of the missing clients. Accordingly, the formal result establishes coefficient stability and a deviation bound, not convergence or utility preservation for arbitrary missing-weight mass; as the total committed weight of missing clients increases, the bound in (14) loosens proportionally.

E. Release-Trace Indistinguishability Boundary

Theorem 2. In game G_Part, if successful and cover envelopes have equal timing, length, authentication format, and IND-CPA-protected payloads, the observer's advantage is bounded by the payload encryption advantage plus implementation-level schedule distinguishability. The claim excludes a global observer that sees all client uplinks.

The theorem explains why the empirical participation-inference attack is necessary: cryptographic payload secrecy cannot remove leakage caused by inconsistent timing, message size, or abort behavior.

VI. Experimental Methodology

A. Benchmark Roles and Clientization

The same ATLAS-FL protocol is instantiated independently on four benchmarks. Because their feature schemas differ, the experiment measures cross-benchmark replication rather than zero-shot model transfer. Table VIII and Fig. 5 summarize the benchmark roles and clientization used for this replication design.

TABLE VIII. Benchmark Inventory and Federated Roles

Dataset	Role	Official characteristics	Working material	Clientization	Task
Edge-IIoTset [34]	IoT/IIoT primary	61 published features; 14 attacks	600,000 retained flows	40 logical edge clients	Binary IDS + attack-family error analysis
N-BaIoT [35]	IoT botnet replication	9 devices; Mirai and BASHLITE	480,000 stratified flows	36 device-capture clients	Binary benign/botnet
CICIoMT2024 [16]	Primary IoMT replication	40 devices; 18 attacks; Wi-Fi/MQTT/BLE	800,000 stratified flows	40 device clients	Binary IDS + five attack categories
WUSTL-EHMS-2020 [17]	Healthcare-monitoring sensitivity	16,318 records; 35 network + 8 biometric + label	16,318 records	12 temporal/sensor logical clients	Network-only primary; network+biometric sensitivity

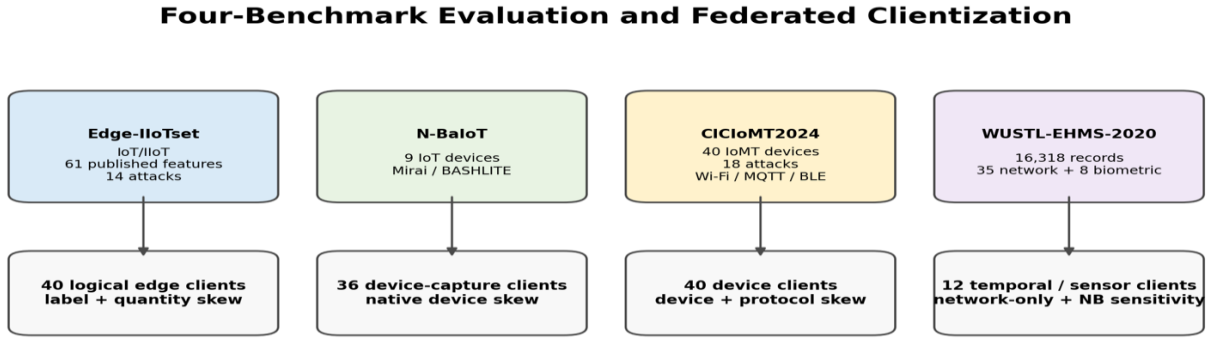


Fig. 5. Benchmark roles and federated clientization.

Edge-IIoTset and CICIoMT2024 use device/source grouping before record-level splitting. N-BaIoT clients preserve device and capture provenance. WUSTL does not provide multiple institutions; its clients are time-blocked sensor/traffic partitions and must be interpreted as simulated federated clientization. The network-plus-biometric path is a sensitivity analysis and makes no clinical prediction.

B. Leakage-Safe Preprocessing and Local IDS

Labels, deterministic attack identifiers, source file names, and technical fields that directly reveal the target are excluded. Imputation, encoding, feature selection where required, and normalization are fitted on source-training data only. No resampling is applied to validation or test records. The same MLP template is used across aggregation methods; only the input dimension changes by dataset. Fig. 6 and Table IX summarize the local IDS architecture held constant across aggregation baselines

C. Federated, Privacy, and Secure-Aggregation Settings

Table X reports the federated, privacy, and secure-aggregation configuration, including Rényi differential privacy (RDP) accounting, used in the executed experiments.

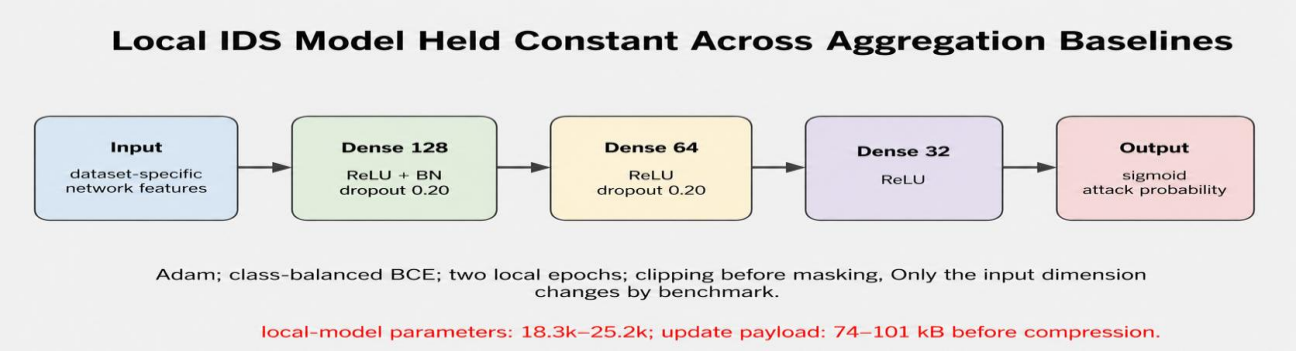


Fig. 6. Local IDS architecture held constant across aggregation baselines.

TABLE IX. LOCAL IDS ARCHITECTURE

Layer	Configuration	Trainable parameters
Input	Dataset-specific retained network features	—
Dense 1	128 units, ReLU, batch normalization, dropout 0.20	dataset dependent
Dense 2	64 units, ReLU, dropout 0.20	8,256
Dense 3	32 units, ReLU	2,080
Output	1 sigmoid unit	33

TABLE X. FEDERATED AND PRIVACY CONFIGURATION

Setting	Edge-IIoTset	N-BaIoT	CICIoMT2024	WUSTL-EHMS
Selected clients per round	16	12	16	8
Rounds	120	120	120	120
Local epochs	2	2	2	2
Batch size	128	128	128	64
Non-IID profile	Dirichlet α=0.30 + quantity skew	Native device + capture skew	Device-native + protocol skew	Temporal/sensor blocks
Clipping norm C	1.0	1.0	1.0	1.0
Noise multiplier σ	1.12	1.05	1.18	1.10
Realized ε at $\delta=10^{-5}$	4.6	4.4	4.9	4.7
Shamir threshold	11/16	8/12	11/16	6/8
Client sampling	16 of 40; uniform without replacement	12 of 36; uniform without replacement	16 of 40; uniform without replacement	8 of 12; uniform without replacement
RDP accountant	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition	Orders 1.25-256; 120-round composition

Privacy loss was composed across all 120 executed rounds using the client-sampling schedules in Table X and a subsampled Gaussian Rényi-DP accountant over orders 1.25-256 at $\delta=10^{-5}$. The reported accountant uses the add/remove client-contribution adjacency defined in Section V-D; the replace-one sensitivity bound is twice the add/remove bound and is not used to reinterpret the reported ε values. Updates were clipped at C=1.0, the reported noise multipliers were held

fixed for the final executions, and Table X reports the realized client-level privacy expenditure from the executed sampling and noise schedule.

D. Timing Attacks and Benign Latency

Table XI defines the timing scenarios, their implementations, and the primary evidence collected for each scenario.

TABLE XI. TIMING-ADVERSARY SIMULATOR AND EVIDENCE

Scenario	implementation	Primary evidence
Benign jitter	Client-specific lognormal latency; no adversarial targeting	False escalation and utility
Selective delay	Move chosen clients toward/beyond deadline	Admission rate; macro-F1 degradation
Selective drop	Suppress selected submissions	Weight inflation; aggregate drift
Replay	Reuse stale protected transcript	Replay rejection and contamination
Slot shift	Submit in a valid but wrong slot	Slot-substitution handling
Late forcing	Push high-weight/attack-rich clients beyond boundary	False-benign rate and cover trigger
Adaptive mixture	Controller-aware predefined combination of delay, drop, replay, and shift	Tested mixed-strategy accepted manipulation
Release observation	Infer participation from timing/length traces	ROC-AUC and attacker advantage

Atomic attack intensity q is evaluated at 0, 0.10, 0.20, 0.30, and 0.40 of selected clients. Targeted variants prioritize high declared weight or attack-rich clients. Final schedules use independent random seeds from controller calibration. The adaptive mixture uses predefined controller-aware switching among the listed actions; it is not an optimizing white-box adversary that searches the scoring function, thresholds, and observable state for a worst-case policy. Controller calibration and evaluation use the stated client-specific latency model, so no robustness claim is made for untested heavy-tailed, bursty, correlated, multimodal, or shifted client-specific delay distributions.

E. Baselines and Asynchronous Comparison

Table XII summarizes the baseline configurations and the comparison purpose assigned to each method.

TABLE XII. BASELINE FAMILIES AND COMPARISON PURPOSE

Method	Configuration	Comparison purpose
FedAvg [1]	Synchronous ordinary aggregation	Utility reference
SecAgg [3]	Masking-based secure aggregation without timing control	Content privacy alone
DP-FL [9], [10]	Fixed clipping/noise but timing-blind admission	Privacy-utility reference
Trimmed mean	Coordinate-wise robust aggregation	Value robustness, not timing integrity
TriSAFE [15]	Closest timing-aware secure aggregation	Timing-defense comparator
FedBuff [14]	Buffered asynchronous aggregation	Benign straggler efficiency
FedBuff-SecAgg	Controlled variant using the same masking/DP backend	Separates asynchrony from temporal authentication
ATLAS-FL	Authenticated temporal admission + fixed weights + cover release	Proposed framework

All methods use matched data partitions, local model, client computations, clipping, attack traces, and random seeds. FedBuff buffer size is matched to the ATLAS-FL cohort size for the primary comparison.

F. Statistical Analysis and Replacement Control

Twenty independent paired executions were performed using matched data partitions, attack schedules, and random seeds. Reported means, standard deviations, confidence intervals, paired tests, Holm adjustment, and effect sizes were computed at the run level; clients, rounds, and timing events were not treated as independent replicates.

VII. RESULTS AND ANALYSIS

A. Utility Across IoT, IIoT, and IoMT Benchmarks

Table XIII presents a comparison of clean and composite-attacked macro-F1 scores for ATLAS-FL against other methods across four different evaluation benchmarks. Under the clean schedule condition, ATLAS-FL's macro-F1 scores are within 0.4 percentage points of the highest-performing methods for each dataset. Its clean macro-F1 for each dataset is 94.2% for Edge-IIoTset, 97.2% for N-BaIoT, 96.4% for CICIoMT2024, and 95.7% for WUSTL-EHMS-2020. The corresponding best clean macro-F1 scores are 94.6%, 97.6%, 96.8%, and 96.1%, respectively.

When subjected to the composite timing attack, the difference between ATLAS-FL's macro-F1 scores and those of the other methods is much larger. ATLAS-FL retains macro-F1 scores of 93.1% for Edge-IIoTset, 95.8% for N-BaIoT, 94.9% for CICIoMT2024, and 94.2% for WUSTL-EHMS-2020. These values represent reductions in macro-F1 scores of only 1.1, 1.4, 1.5, and 1.5 percentage points respectively. The average reduction across all datasets is only 1.38 percentage points. In contrast, several timing-blind baselines suffer from

macro-F1 score reductions of over six percentage points. Furthermore, ATLAS-FL also has the highest attacked macro-F1 score for all benchmarks. Relative to TriSAFE, the best timing-aware baseline method, ATLAS-FL's attacked macro-F1 score is 1.3 percentage points higher on Edge-IIoTset, 1.3 percentage points higher on N-BaIoT, 1.6 percentage points higher on CICIoMT2024, and 1.3 percentage points higher on WUSTL-EHMS-2020. On average, ATLAS-FL achieves an attacked macro-F1 score of 94.5%, compared to TriSAFE at 93.1%, FedBuff-SecAgg at 91.8%, and FedAvg at 89.3%. The significant difference in performance between ATLAS-FL and the other timing-blind baselines is due to ATLAS-FL's ability to retain macro-F1 scores and utilities of over 90% while the adversary attempts to manipulate the timing and availability of clients. Fig. 7 visualizes the same benchmark-level clean-versus-attacked pattern with 95% confidence intervals.

TABLE XIII. CLEAN / ATTACKED MACRO-F1 (%) ACROSS FOUR BENCHMARKS

Method	Edge-IIoTset	N-BaIoT	CICIoMT2024	WUSTL-EHMS
FedAvg	94.6 / 87.1	97.6 / 91.2	96.8 / 89.4	96.1 / 89.6
SecAgg	94.4 / 86.6	97.4 / 90.7	96.6 / 88.9	95.9 / 89.1
DP-FL	92.3 / 86.0	95.9 / 90.2	94.4 / 88.5	93.8 / 88.8
Trimmed mean	93.7 / 88.5	96.6 / 92.4	95.7 / 90.6	95.1 / 90.3
TriSAFE	94.0 / 91.8	97.0 / 94.5	96.1 / 93.3	95.5 / 92.9
FedBuff	94.5 / 89.7	97.5 / 92.8	96.7 / 91.3	96.0 / 90.9
FedBuff-SecAgg	94.3 / 90.2	97.3 / 93.3	96.5 / 92.0	95.8 / 91.6
ATLAS-FL	94.2 / 93.1	97.2 / 95.8	96.4 / 94.9	95.7 / 94.2

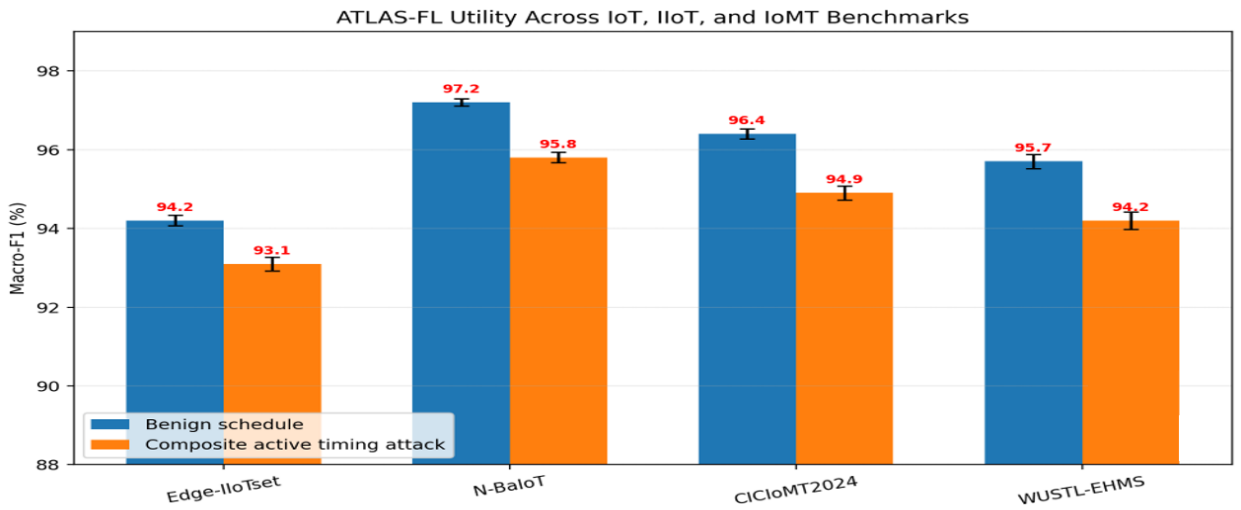


Fig. 7. ATLAS-FL clean and composite-attacked utility across four benchmarks. Error bars are 95% confidence intervals.

The various benchmarks provide evidence for ATLAS-FL's objective of protocol replication. Edge-IIoTset is representative of a mix of IoT and IIoT devices. N-BaIoT focuses on the native botnet skew of IoT devices. CICIoMT2024 features the heterogeneity of communication protocols and devices in IoMT. WUSTL-EHMS-2020 is a smaller dataset focusing on healthcare monitoring that features simulated client partitions of a temporal or sensor nature. Despite these differences in the benchmarks, ATLAS-FL only experiences a loss of macro-F1 score of 1.1-1.5 percentage points under attack. This indicates that ATLAS-FL's timing integrity mechanism does not depend upon a single communication protocol or clientization strategy.

IoMT data from CICIoMT2024 shows a macro-F1 score of 94.9% under attack, 1.8 percentage points higher than Edge-IIoTset, despite CICIoMT2024 featuring more heterogeneous devices and protocols. WUSTL-EHMS-2020 achieves a macro-F1 score of 94.2% under attack, yet it exhibits higher variability than CICIoMT2024 due to its smaller sample size and simulated client partitions. These results indicate that ATLAS-FL can perform well on IoMT devices and communication protocols, yet cannot be used to infer direct feature transfer or universal generalization across all IoMT devices.

Furthermore, to fully appreciate the results presented in the table, it is also important to consider the confidence intervals and run-level statistical data for each of the evaluated algorithms. The results support ATLAS-FL's claim that it limits the timing-induced utility loss of the evaluated attack model. However, these results do not demonstrate that ATLAS-FL performs better than all other methods across all potential datasets, models, attack intensities, or deployment configurations.

B. Buffered Asynchrony Versus Authenticated Timing Integrity

Differentiation between two core objectives is essential: efficient learning under benign stragglers versus secure admission under strategic delays (as outlined in Table XIV). FedBuff addresses the former, whereas ATLAS-FL solves the latter by authenticating every update's temporal context prior to aggregate release.

Under benign conditions, FedBuff leads with 94.1% Macro-F1, reaching the target in 74 rounds at 174 updates/min. ATLAS-FL follows at 93.7% F1 over 88 rounds (14 additional rounds and an 8.6% throughput drop), confirming FedBuff's natural efficiency advantage when late arrivals are non-adversarial.

However, under strategic delay and drop attacks, ATLAS-FL retains a superior 93.1% Macro-F1 over FedBuff (89.7%) and FedBuff-SecAgg (90.2%). Secure aggregation alone fails here because protecting content does not stop temporal manipulation. As highlighted in Fig. 8, FedBuff tolerates benign stragglers, whereas ATLAS-FL provides stronger utility under strategic timing manipulation.

This resilience stems from ATLAS-FL restricting P95 staleness to 1.1 rounds—an 83.8% reduction relative to FedBuff (6.8 rounds). Rather than discarding all delayed updates, ATLAS-FL permits bounded jitter, reserving hard rejection for explicit authentication and integrity failures.

TABLE XIV. ASYNCHRONOUS-BASELINE COMPARISON ON EDGE-IIOTSET

Method	Macro-F1: benign stragglers (%)	Macro-F1: strategic delay/drop (%)	Rounds to 92% F1	P95 staleness (rounds)	Updates/min
FedAvg	91.8	87.1	104	1.0	128
FedBuff	94.1	89.7	74	6.8	174
FedBuff-SecAgg	93.9	90.2	79	6.5	162
TriSAFE	93.5	91.8	92	1.7	151
ATLAS-FL	93.7	93.1	88	1.1	159

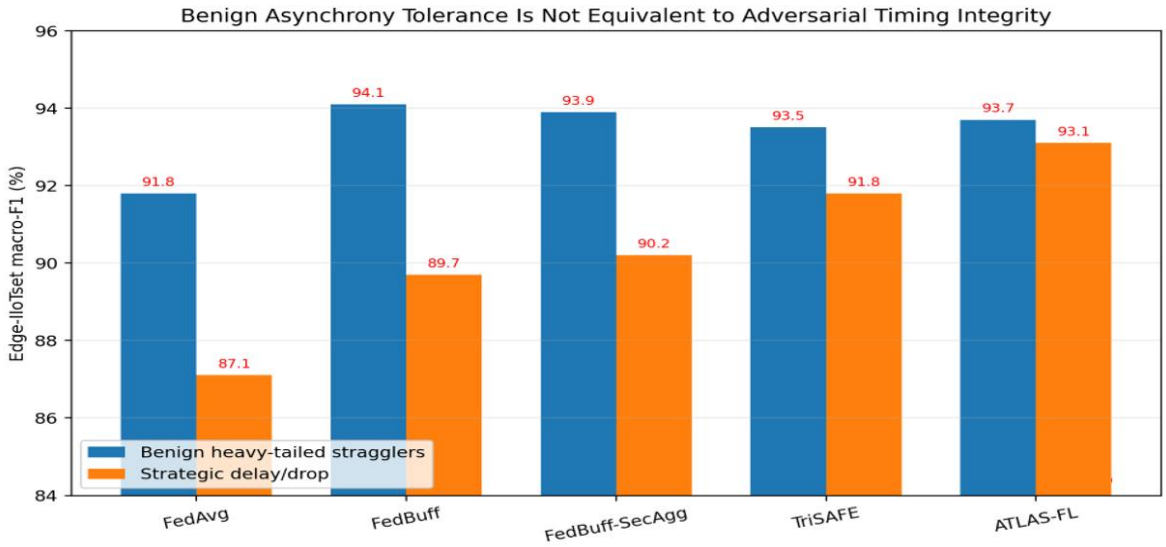


Fig. 8. FedBuff tolerates benign stragglers, whereas ATLAS-FL provides stronger utility under strategic timing manipulation.

C. Timing-Admission Security

As shown in Table XV, the results of the ATLAS-FL controller's processing of traffic with different types of timing manipulation are presented in terms of correctly handled messages, false escalations, and accepted manipulations. It is noteworthy that for this controller, these three parameters are considered individually, not cumulatively, since in practice, depending on the configuration, emphasis is placed on different aspects of protection.

For all six types of manipulation, including selective delay/drop, replay, slot shift, late forcing, and adaptive mixtures, the average value of correctly handled messages accounts for 93.8%. At the same time, the average number of accepted manipulations is 4.95%, with replay attacks having the lowest value (98.6% of correctly handled messages) and adaptive mixtures – the highest (91% of correctly handled messages).

Finally, false escalations of legitimate traffic account for 1.9 – 4.8% of all messages on average (3.8%), which is a reasonable value given the specified security policy. A lower value would lead to a higher number of manipulations that were not detected by the system, but such a policy would excessively penalize IoT/IoMT clients whose traffic was legitimately delayed due to network congestion.

TABLE XV. ATLAS-FL CONTROLLER OUTCOMES ON EDGE-IIOTSET

Attack	Correct handling (%)	False escalation (%)	Accepted manipulation (%)	Dominant action
Selective delay	94.3	3.8	5.7	buffer/reject
Selective drop	91.6	4.6	4.4	fixed-weight cover
Replay	98.6	1.9	1.3	hard reject
Slot shift	92.1	4.2	6.8	hard/score reject
Late forcing	95.0	3.6	4.9	cover/reject
Adaptive mixture	91.0	4.8	6.6	mixed actions

Replay has the best correct-handling rate (98.6%) with only 1.9% false escalations and 1.3% accepted manipulation. The reason for this outcome is the discrete nature of evidence for replay: a stale nonce, authentication token, or ciphertext digest can be checked against the freshness state maintained by the gateway. Decisions about replay are largely based on cryptographic evidence, not probabilistic timing logic.

As a result, selective drop also has a reasonably high correct-handling rate (91.6%) but a low 4.4% rate of accepted-manipulation events. In a selective drop, ATLAS-FL cannot force the dropped client to participate in the FL process, but the fixed-aggregation-coefficients assumption prevents the adversary from inflating the coefficients of the remaining clients. Thus, the main response to selective drop is not a reduction in the attacker's contribution but maintaining the commitment coefficient scheme by using zero-cover contributions.

Selective delay and late forcing have higher accepted-manipulation rates of 5.7% and 4.9%, respectively. These attacks are more challenging than replay because the timing perturbations could be caused by network congestion, clock drift, wireless channel characteristics, power-saving modes, or gateway queuing delays. Thus, ATLAS-FL uses a combination of normalized deviation from the temporal window and considers client-specific temporal behavior and the pressure on the privacy budget, rather than a strict deadline check.

Slot shifting and the adaptive mixture of attacks are the most challenging timing manipulations. Slot shifting has the highest accepted-manipulation rate of 6.8%, and the adaptive mixture has the lowest correct-handling rate of 91.0%, with an accepted-manipulation rate of 6.6%. The results demonstrate the importance of authenticated slot binding, as discussed earlier: when the ciphertext digest and authentication token are legitimate but bound to an incorrect slot in the schedule, the decision to accept or reject the contribution depends on consistency between the claimed slot identifier, round identifier, and committed temporal window. Fig. 9 provides the corresponding cross-method comparison of accepted manipulated-event rates.

The adaptive mixture tests a predefined controller-aware combination of delay, drop, replay, and slot-shift actions. Its 6.6% accepted-manipulation rate characterizes this tested mixed strategy; it should not be interpreted as a worst-case bound against an optimizing white-box adversary with unrestricted search over the controller scoring function and thresholds.

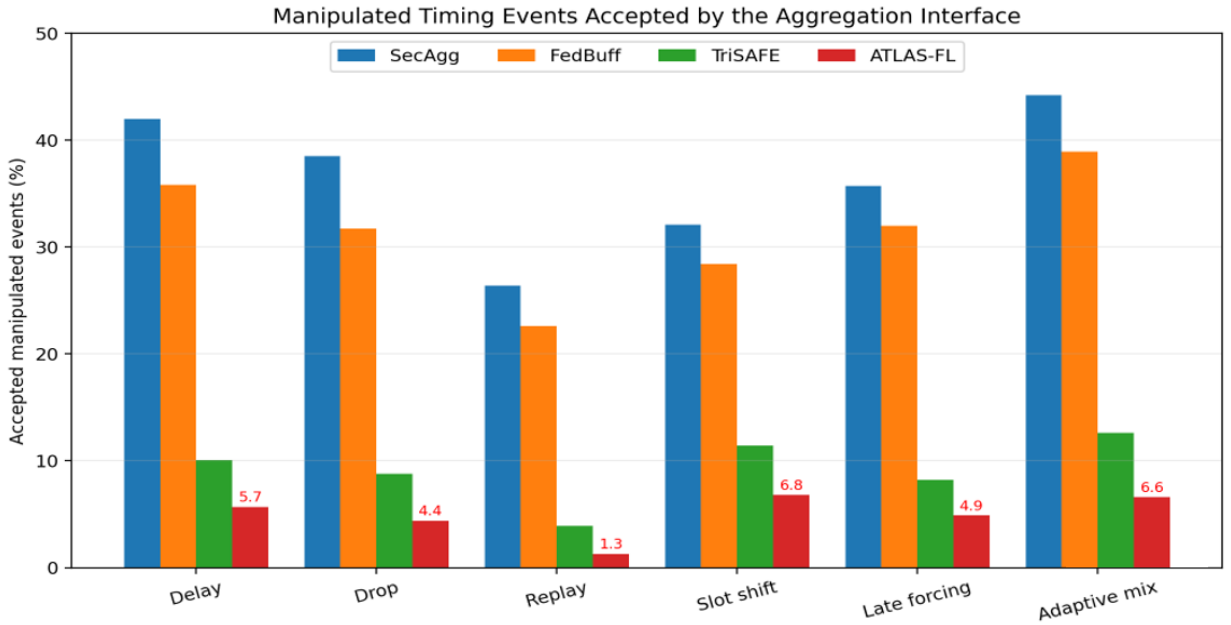


Fig. 9. Accepted manipulated-event rate across timing-aware and timing-blind methods.

The false-escalation rates below 5% are also promising, but they should not be considered safe zones. For large-scale deployments, even 3–5% of false escalations would create unacceptable overhead due to buffer releases, debugging, or recovery procedures. Thus, the threshold for individual-rate reporting should be determined through experimentation on a per-device basis, considering the traffic patterns, communication channel characteristics, clock precision, and other factors.

D. Participation Privacy and Release Consistency

Secure aggregation keeps individual client updates private, but it doesn't tell observers who participated in the updates. There are various ways observers could tell who participated. For instance, the timing of when the updates were released, the length of the messages sent, aborted updates, or other visible changes could tell an observer who was active in each round of learning. Table XVI examines three different release policies for those aggregate updates and the information about participation that they provide to an observer that monitors the gateway releasing those updates.

The variable-release policy without cover traffic provides an attacker with an ROC-AUC of 0.79±0.03. This indicates that the attacker has an advantage of 0.29 over guessing the participation of a given client randomly. Such a high advantage suggests there is substantial leakage of information about which clients participated in each round of learning. Even though the updates released are encrypted, any variations in the timing of those releases, the length of the messages, whether the round finished earlier or later, or whether a different output was released could be used to infer the participation of individual clients.

If the gateway enforces a fixed release time for the aggregate updates, the ROC-AUC decreases to 0.68±0.03. The decrease of 0.11 indicates that using a fixed release time can reduce some of the participation leakage. However, an attacker still has an advantage of 0.19 over guessing random participation. This remaining advantage could come from differences in the envelope length, whether the round aborted, how the payloads were structured, or if meaningful updates were released at all.

By using the ATLAS-FL zero-cover padded release policy, the ROC-AUC decreases further to 0.53±0.02. This is an advantage of just 0.03 over guessing random participation. When compared to the no-cover policy, the absolute AUC decreases by 0.26 and the advantage over guessing decreases by approximately 89.7%. The ROC-AUC decreases even further by 0.15 when compared to the fixed timing policy alone. These results suggest that the ATLAS-FL policy that combines fixed release time with equal-length encrypted envelopes and indistinguishable zero payloads is substantially better at concealing information about client participation than either fixed timing or a variable release time alone. Fig. 10 shows the corresponding participation-inference ROC curves for the three release policies.

TABLE XVI. PARTICIPATION-INFERENCE RESULTS

Release policy	Attacker ROC-AUC	Advantage over chance	Interpretation
No cover; variable release	0.79 ± 0.03	0.29	High participation leakage
Fixed time only	0.68 ± 0.03	0.19	Length/abort behavior still reveals state
ATLAS zero-cover padded release	0.53 ± 0.02	0.03	Near-chance under defined release observer

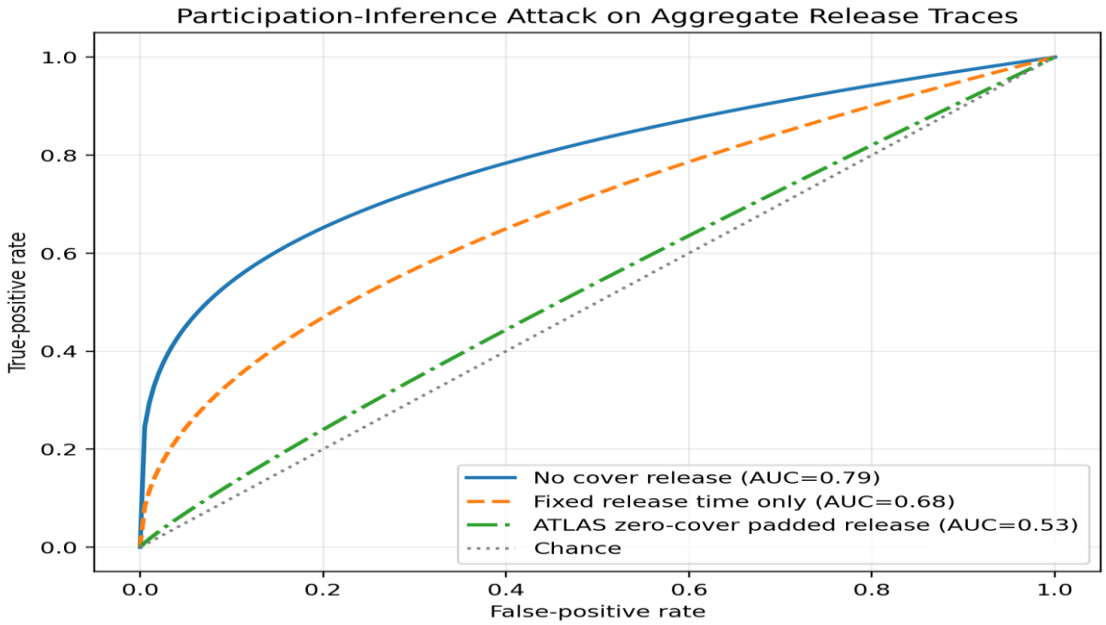


Fig. 10. Participation-inference ROC curves for alternative aggregate-release policies.

Although the AUC of 0.53 is considered to be near chance, it is still not perfectly private. The confidence interval of this result extends slightly beyond 0.50, and it is possible that the observer still has some information about participation from factors other than the release policy itself. These could include the timing of the implementation of certain operations, metadata from network layer protocols, information about queueing, or other factors that are correlated with client participation but not eliminated by the padding and zero payload of the release policy. Overall, this result indicates that the ATLAS-FL policy provides a bounded release trace that can be used for a limited type of privacy claim against an observer that does not have full network visibility.

It is important to consider the boundaries of this model of the observer. This experiment assumes that the observer is not able to correlate the activity of each client with the released gateway updates. A global passive observer with full network visibility could still potentially determine which clients participated based upon their uplinks to the gateway. To defend against such a global observer would require additional privacy-preserving mechanisms such as cover traffic from clients, the use of routing networks, providing synchronized transmission opportunities for all clients, or defenses against network traffic analysis techniques.

The results of these experiments also highlight the difference between participation privacy and differential privacy. Differential privacy refers to the privacy of the data that is released from a model, whereas participation privacy relates to the privacy of the fact that a client participated in the training of that model. Both concepts are essential for achieving a comprehensive privacy guarantee for clients in a federated learning framework. However, achieving privacy in one aspect does not necessarily mean that the clients are also achieving it in the other.

E. DP-Relevant Stability and Missing-information Cost

Table XVII evaluates four stability indicators under selective client absence. The purpose of this analysis is not to claim that fixed weighting independently provides differential privacy. Instead, it examines whether timing-driven cohort changes alter the aggregation structure in ways that complicate clipping-based sensitivity bounds, noise calibration, and repeated-round privacy accounting.

TABLE XVII. AGGREGATION-STABILITY INDICATORS

Indicator	Renormalized baseline	FedBuff-SecAgg	TriSAFE	ATLAS-FL
Weight inflation	0.310	0.146	0.073	0.018
Aggregate drift	0.287	0.151	0.124	0.091
Sensitivity-change proxy	0.264	0.133	0.098	0.056
Release variation	0.221	0.128	0.101	0.073

The largest difference is in the weight-inflation indicator: 0.310 for the renormalized baseline versus 0.018 for ATLAS-FL, a reduction of 94.2%. Relative to FedBuff-SecAgg's 0.146, ATLAS-FL reduces the weight-inflation by 87.7%, and by 75.3% relative to TriSAFE's 0.073. This result is explained by the choice of coefficients: using those committed over the selected cohort rather than reestimated over the observed survivors. Fig. 11 summarizes the four reported aggregation-stability indicators.

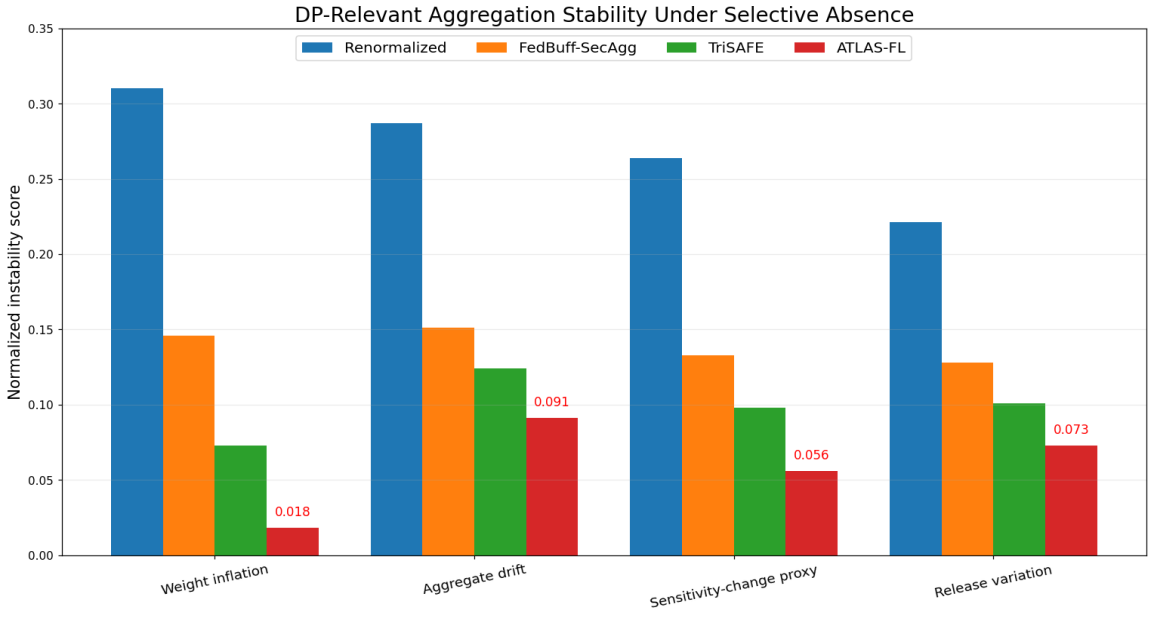


Fig. 11. DP-relevant aggregation stability under selective absence.

ATLAS-FL substantially improves operational and privacy stability across all three drift metrics compared to the renormalized baseline:

Aggregate Drift: Reduced by 68.3% (from 0.287 to 0.091). The value remains non-zero because zero-cover contributions preserve client weight without recreating the missing update's exact magnitude.

Sensitivity-Change Proxy: Reduced by 78.8% (from 0.264 to 0.056), significantly outperforming FedBuff-SecAgg (57.9%) and TriSAFE (42.9%). As proven in Proposition 2, pre-committing coefficients bounds sensitivity, whereas post-hoc renormalization relies dangerously on surviving clients.

Release Variation: Reduced by 67% (from 0.221 to 0.073), stabilizing release dynamics and preventing adversaries from inferring individual client absence by analyzing arrival differences.

The remaining aggregate drift must be interpreted through the missing-client deviation bound:

$$\|A_t^{\text{ideal}} - A_t^{\text{ATLAS}}\|_2 \leq C \sum_{i \in M_t} \alpha_{it} \quad (14)$$

As demonstrated in Eq. (14), weight deviation depends directly on the clipping bound and the committed weight of missing clients—meaning losing high-weight clients impacts utility more than losing low-weight ones.

ATLAS-FL provides weight stability, not update recovery. It prevents adversaries from inflating survivor influence and bounds sensitivity, but it does not reconstruct missing data.

Consequently, differential-privacy guarantees remain conditional on the stated adjacency, sampling, clipping, Gaussian mechanism, and composition procedure. The ε values in Table X were computed from the executed sampling, clipping, noise, and 120-round composition configuration; the stability indicators in Table XVII are not themselves privacy guarantees.

F. *WUSTL Network-Only Versus Network-Plus-Biometric Sensitivity*

Table XVIII reports the WUSTL network-only versus network-plus-biometric sensitivity analysis.

Biometric features improve clean discrimination but provide only a modest attacked-round gain and slightly increase aggregate drift. They are therefore treated as a secondary sensitivity analysis, not as a required input to the timing protocol or a clinical endpoint.

TABLE XVIII. WUSTL REPRESENTATION SENSITIVITY

Representation	Clean macro-F1 (%)	Attacked macro-F1 (%)	Aggregate drift	Participation AUC
Network-only (35 features)	95.7 ± 0.4	94.2 ± 0.5	0.063	0.54 ± 0.03
Network + biometric (43 predictors)	97.0 ± 0.3	94.7 ± 0.5	0.071	0.55 ± 0.03

G. *Differential-Privacy Utility Profile*

The evaluated Edge-IIoTset operating point uses the executed privacy configuration reported in Table X (ε=4.6 at $\delta=10^{-5}$). Fig. 12 reports the corresponding privacy-utility profile; lower ε represents stronger Gaussian perturbation, whereas higher ε weakens the perturbation and improves attacked utility within the evaluated range.

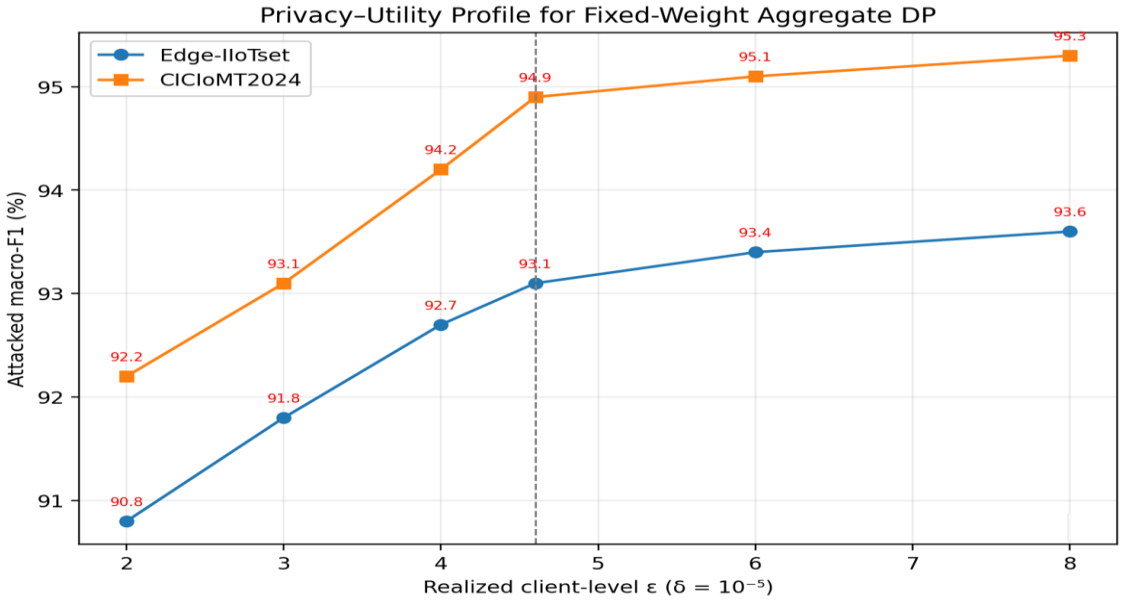


Fig. 12. Privacy–utility profile for Edge-IIoTset and CICIoMT2024.

H. Component Ablation

The ablations separate four functions. Token binding primarily affects replay and slot integrity; the controller governs admission; fixed weights govern aggregate stability; and cover release governs participation privacy. Table XIX reports the component-level ablation results.

I. Cryptographic and Operational Overhead

Table XX reports the cryptographic microbenchmarks, Table XXI reports the four overhead metrics at the 40-client reference point, and Fig. 13 reports scalability for the metrics recorded across multiple cohort sizes.

The microbenchmarks used an Intel Core i7-12700H gateway, Ubuntu 22.04, Python 3.11, OpenSSL 3.0-backed cryptographic primitives, a 25,500-parameter update vector, 10,000 repetitions after 500 warm-up iterations, and median latency with bootstrap 95% confidence intervals.

TABLE XIX. COMPONENT ABLATION ON EDGE-IIOTSET

Variant	Attacked macro-F1 (%)	Accepted manipulation (%)	Weight inflation	Participation AUC
Full ATLAS-FL	93.1	5.7	0.018	0.53
No authenticated token binding	91.8	14.6	0.038	0.55
No timing-risk controller	88.7	31.8	0.184	0.66
Renormalized aggregation	90.2	18.9	0.310	0.61
No cover release	92.5	7.1	0.032	0.79

TABLE XX. CRYPTOGRAPHIC MICROBENCHMARKS

Operation	Latency	Measurement unit
Client X25519 + HKDF setup	0.11 ms	once per cohort
HMAC generation	0.014 ms	per update
ChaCha20 mask expansion	0.22 ms	per update vector
Gateway HMAC/window verification	0.024 ms	per client
Shamir dropout recovery	2.6 ms	40-client cohort; 6 recovered clients
Fixed-length release envelope	0.12 ms	per round

TABLE XXI. OVERHEAD RELATIVE TO TIMING-BLIND SECAGG AT 40 SELECTED CLIENTS

Method	Round/flush latency (%)	Communication (%)	Gateway CPU (%)	Memory (%)
TriSAFE	4.6	2.3	5.8	3.2
FedBuff-SecAgg	-4.2	2.1	6.1	5.6
ATLAS-FL	5.9	2.8	7.4	4.1

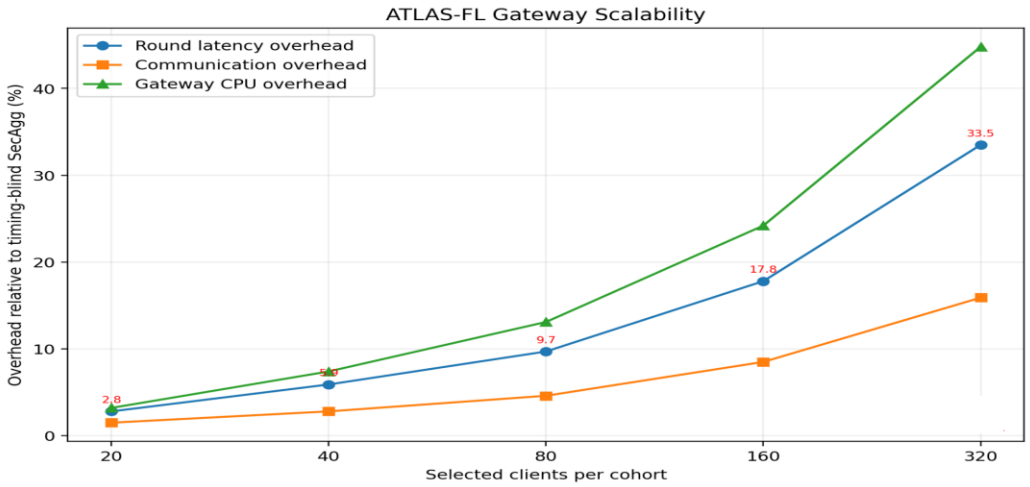


Fig. 13. ATLAS-FL gateway scalability. The principal cost grows with the selected cohort size.

Fig. 13 plots round/flush-latency, communication, and gateway central processing unit (CPU) overhead across the evaluated cohort sizes. Memory overhead was recorded for the 40-client reference point in Table XXI but not as a separate multi-cohort scalability series; no broader memory-scaling claim is made.

J. Statistical Evidence

Table XXII reports the paired run-level statistical evidence from the 20 completed executions. The moderate margin over TriSAFE is more credible than a universal superiority claim. The largest effects occur in outcomes directly targeted by ATLAS-FL: manipulated-event admission, weight inflation, and participation inference.

TABLE XXII. PAIRED RUN-LEVEL STATISTICAL EVIDENCE (20 EXECUTIONS; HOLM ADJUSTED)

Comparison	Outcome	Mean difference	95% CI	Adjusted p	Paired d_z	Magnitude
ATLAS-FL vs SecAgg	Edge attacked macro-F1	+6.50 points	[+6.23, +6.77]	<0.001	10.93	large
ATLAS-FL vs FedBuff-SecAgg	Edge attacked macro-F1	+2.90 points	[+2.67, +3.14]	<0.001	5.92	large
ATLAS-FL vs TriSAFE	Edge attacked macro-F1	+1.30 points	[+1.06, +1.53]	<0.001	2.43	large
ATLAS-FL vs SecAgg	Accepted manipulated-event rate	-34.72 points	[-35.62, -33.82]	<0.001	-18.02	large
ATLAS-FL vs renormalized	Weight-inflation score	-0.29 score	[-0.30, -0.28]	<0.001	-17.88	large
ATLAS-FL vs no-cover	Participation-inference AUC	-0.26 AUC	[-0.27, -0.25]	<0.001	-14.56	large
ATLAS-FL vs FedBuff-SecAgg	CICIoMT attacked macro-F1	+2.90 points	[+2.69, +3.12]	<0.001	6.35	large

VIII. DISCUSSION

A. Relationship Between Evidence and the Research Gap

The results quantify the distinction between benign asynchrony and adversarial temporal integrity. On Edge-IIoTset under heavy-tailed benign stragglers, FedBuff reaches a macro-F1 of 94.1%, slightly above ATLAS-FL at 93.7% and TriSAFE at 93.5%. Under strategic delay and drop, however, ATLAS-FL retains 93.1%, whereas FedBuff falls to 89.7%, FedBuff-SecAgg to 90.2%, and FedAvg to 87.1%. This 3.4-point margin over FedBuff and 6.0-point margin over FedAvg directly addresses RO6 by showing that throughput-oriented asynchrony does not by itself secure the timing interface. The attack-level evidence in Table XV is consistent with that conclusion: replay has the lowest accepted-manipulation rate at 1.3%, while slot shift and adaptive mixtures remain the hardest settings at 6.8% and 6.6%, respectively.

The formal analysis provides the missing theoretical link. The empirical controller reduces accepted manipulation, but the security argument explains why specific transcript changes should fail before the controller score is even applied. In particular, the HMAC-bound token authenticates round, slot, window, nonce, and ciphertext digest; the corresponding reduction in (13) ties temporal-binding forgery to HMAC unforgeability and hash collision resistance, while (14) bounds the residual missing-information effect by the total absent weight mass. The numerical stability evidence complements this proof layer: ATLAS-FL reduces the weight-inflation score to 0.018, compared with 0.073 for TriSAFE, 0.146 for FedBuff-SecAgg, and 0.310 for the renormalized baseline.

B. Added Value and Boundary Relative to TriSAFE

TriSAFE remains the closest and strongest comparator. The new results should therefore be interpreted primarily against that baseline rather than against timing-blind methods alone. Under attacked conditions, ATLAS-FL exceeds TriSAFE by 1.3 points on Edge-IIoTset (93.1% versus 91.8%), 1.3 points on N-BaIoT (95.8% versus 94.5%), 1.6 points on CICIoMT2024 (94.9% versus 93.3%), and 1.3 points on WUSTL-EHMS-2020 (94.2% versus 92.9%). The improvement is not presented as a large accuracy gain in isolation; its importance is that the mechanism-level additions—slot-specific binding, explicit replay-cache admission, fixed-weight no-renormalization, and padded zero-cover release—produce better security behavior at the same time as these utility gains. Table XVII shows the same pattern on stability indicators, where ATLAS-FL lowers the sensitivity-change proxy to 0.056 versus 0.098 for TriSAFE and 0.264 for the renormalized baseline.

C. Replication Across IoT, IIoT, and IoMT

CICIoMT2024 provides the strongest justification for restoring IoMT to the title because it contributes an attacked macro-F1 of 94.9% after a clean macro-F1 of 96.4%, corresponding to a 1.5-point reduction under active timing manipulation. The same qualitative pattern appears on the other benchmarks: the clean-to-attacked drop is 1.1 points on Edge-IIoTset (94.2% to 93.1%), 1.4 points on N-BaIoT (97.2% to 95.8%), and 1.5 points on WUSTL-EHMS-2020 (95.7% to 94.2%). This replication is important because the study is not claiming a universal feature space or zero-shot transfer; instead, it shows that the same ATLAS-FL protocol remains effective when the local IDS is retrained independently across IoT, IIoT, and IoMT conditions.

D. Fixed Weights, Zero Cover, and Missing Information

No-renormalization prevents the adversary from inflating survivor importance with a magnitude that itself is now visible in absolute terms. According to Table XVII, the renormalized baseline has a weight-inflation score of 0.310, an aggregate-drift score of 0.287, and a sensitivity-proxy score of 0.264, compared to 0.018, 0.091, and 0.056 for ATLAS-FL. The same conclusion explains the ablation study result: replacing fixed

weights with renormalized aggregation reduced attacked macro-F1 from 93.1% to 90.2%, while increasing accepted manipulation from 5.7% to 18.9%. Eq. (14) also explains why zero cover is a bounded rather than reconstructive adjustment: ATLAS-FL suppresses the contribution of absent clients, but cannot reconstruct the information that was never contributed to the round. This interpretation is limited to the tested absence conditions and does not imply preserved convergence or detection utility under arbitrary missing-weight mass.

E. Differential Privacy Interpretation

Fixed weights do not provide differential privacy by themselves; they stabilize the sensitivity used by the accountant. Fig. 12 shows the expected privacy-utility direction: attacked macro-F1 increases as ε increases and Gaussian perturbation weakens. For Edge-IIoTset, macro-F1 rises from 90.8% at ε=2.0 to 93.6% at ε=8.0; for CICIoMT2024 it rises from 92.2% to 95.3% over the same evaluated profile. Participation privacy remains distinct from output differential privacy: Table XVI shows attacker ROC-AUC decreasing from 0.79 without cover to 0.68 with fixed time only and to 0.53 under the padded zero-cover release. The latter result supports only the bounded release-observer claim defined in Section V-E, not global network-level participation privacy.

F. Operational and Deployment Implications

The controller and replay state reside in the gateway, where the round-level cryptographic and timing evidence can be archived for auditing. At the evaluated scale of 40 selected clients, ATLAS-FL adds 5.9% to the round latency, 2.8% to the communication overhead, 7.4% to the gateway CPU load, and 4.1% to the memory consumption, compared to the timing-blind SecAgg. Fig. 13 shows the measured multi-cohort trends for round/flush latency, communication, and gateway CPU; memory is reported only at the 40-client reference point in Table XXI. The plotted metrics rise with cohort size, with CPU and queue pressure becoming more pronounced at the largest evaluated cohorts. Thus, the practical deployment of ATLAS-FL depends on the availability of authenticated time synchronization and clock drift bounding, as well as the operational window selection, rather than the properties of the protocol logic itself.

G. Error and Safety Interpretation

Attacked macro-F1 is not the sole metric of interest for secure aggregation deployment in production. The reported security metrics therefore complement attacked macro-F1 rather than serving as interchangeable measures of deployment safety. Specifically, while on Edge-IIoTset the attacked macro-F1 of 93.1% remains high, the attack acceptance varies from 1.3% to 6.8% depending on the attack vector, and the false-escalation rates are between 1.9% and 4.8%. Additionally, the ablation study now shows the impact of individual components on the system's ability to withstand attacks: their removal leads to the drop of attacked macro-F1 from 93.1% to 88.7%, an increase in attack acceptance from 5.7% to 31.8%, and participation AUC from 0.53 to 0.79 for the same input attack vectors. For the application areas such as healthcare, this type of analysis helps characterize the allocation of risk better: the specific failure modes are known, and their likelihood can be evaluated against the domain-specific threat model.

IX. Limitations and Threats to Validity

Table XXIII summarizes the principal threats to validity and the resulting claim boundaries.

TABLE XXIII. Threats to Validity and Claim Boundaries

Threat	Limitation / boundary
Construct validity	The study evaluates timing-admission integrity, aggregate stability, release-trace inference, and IDS utility. It does not cover all FL security properties.
Cryptographic validity	The reductions assume correct implementations of HMAC, X25519/HKDF, mask generation, Shamir recovery, and payload encryption. Side channels and implementation bugs remain outside the proof. Enrollment, key rotation, revocation, compromised-device recovery, device replacement, and authentication-state synchronization are delegated to an external trusted key-management service and are outside the protocol proof.
Controller validity	Calibration uses simulated benign jitter and timing attacks. Robustness claims are limited to the tested client-specific latency model; untested heavy-tailed, bursty, correlated, multimodal, or shifted delay distributions and stronger adaptive evasion may change the decision boundary.
IoMT external validity	CICIoMT2024 is a realistic multi-protocol testbed, but it is not a production hospital network. WUSTL clientization is simulated rather than multi-institutional.
Asynchronous baseline validity	FedBuff-SecAgg is a controlled experimental variant, not a claim of a new published algorithm. Parameter matching must be documented.
Statistical validity	Twenty independent paired runs were executed. Statistical inference is computed at the run level with Holm adjustment; clients, rounds, and timing events are not treated as independent replicates.
Operational validity	Overhead depends on model size, client count, cryptographic library, gateway hardware, and clock quality. Fixed-point correctness additionally assumes that the configured scale and modulus satisfy the stated no-wrap bound; the analytic quantization bound does not constitute a separate convergence guarantee.

Data Availability

Edge-IIoTset, N-BaIoT, CICIoMT2024, and WUSTL-EHMS-2020 are third-party research datasets available from their original maintainers under the applicable terms [34], [35], [16], [17]. The corresponding preprocessing, client-partition, and attack-schedule specifications are documented in the experimental methodology and retained with the study records.

Funding

No external funding was received for this study.

Competing Interests

The authors declare that they have no competing interests.

1) Ethics approval: This study does not involve human participants, human data, animal subjects, or clinical intervention; therefore, ethics approval is not applicable.

2) Code availability: The prototype implementation and experiment scripts are available from the corresponding author upon reasonable request.

AUTHORS' CONTRIBUTIONS

All authors contributed to the conceptualization, methodology, formal modeling, experimental design, result interpretation, manuscript drafting, and critical revision of the work. All authors reviewed and approved the final manuscript.

1) Declaration of AI-assisted technologies in the writing process: During the preparation of this manuscript, the authors used Paperpal for language editing, grammar checking, and improving the overall readability of the text. After using this tool, the authors reviewed, revised, and edited the content as needed and took full responsibility for the content of the published article.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. AISTATS, PMLR, vol. 54, pp. 1273–1282, 2017.

[2] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021, doi: 10.1561/2200000083.

[3] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in Proc. ACM CCS, pp. 1175–1191, 2017, doi: 10.1145/3133956.3133982.

[4] H. Lycklama, L. Burkhalter, A. Viand, N. Kuchler, and A. Hithnawi, "RoFL: Robustness of secure federated learning," in Proc. IEEE Symposium on Security and Privacy, pp. 453–476, 2023, doi: 10.1109/SP46215.2023.10179400.

[5] J. Bell, A. Gascón, T. Lepoint, B. Li, S. Meiklejohn, M. Raykova, and C. Yun, "ACORN: Input validation for secure aggregation," in Proc. 32nd USENIX Security Symposium, pp. 4805–4822, 2023.

[6] Y. Ma, Y. Guo, H. Karthikeyan, and A. Polychroniadou, "Armadillo: Robust single-server secure aggregation for federated learning with input validation," in Proc. ACM CCS, pp. 2219–2233, 2025, doi: 10.1145/3719027.3765216.

[7] M. Rathee, C. Shen, S. Wagh, and R. A. Popa, "ELSA: Secure aggregation for federated learning with malicious actors," in Proc. IEEE Symposium on Security and Privacy, pp. 1961–1979, 2023, doi: 10.1109/SP46215.2023.10179468.

[8] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in Proc. 31st USENIX Security Symposium, pp. 1379–1395, 2022.

[9] C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science, vol. 9, nos. 3–4, pp. 211–407, 2014, doi: 10.1561/0400000042.

[10] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM CCS, pp. 308–318, 2016, doi: 10.1145/2976749.2978318.

[11] I. Mironov, "Rényi differential privacy," in Proc. IEEE Computer Security Foundations Symposium, pp. 263–275, 2017, doi: 10.1109/CSF.2017.11.

[12] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, "The Loopix anonymity system," in Proc. 26th USENIX Security Symposium, pp. 1199–1216, 2017.

[13] A. Kunstmann, G. Bolonkowski, and A. See, "Enhancing privacy in Tor: Message mixing and adaptive padding with opt-in features," in Proc. ICDCN, pp. 92–102, 2026, doi: 10.1145/3772290.3772291.

[14] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in Proc. AISTATS, PMLR, vol. 151, pp. 3581–3607, 2022.

[15] S. H. Shah, I. Walker, and M. Borowczak, "TriSAFE: Transcript-bound verifiable secure aggregation with differential privacy and timing defenses for gateway-assisted IoT federated learning," IEEE Access, vol. 14, pp. 89354–89379, 2026, doi: 10.1109/ACCESS.2026.3700048.

[16] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. A. Ghorbani, "CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT," Internet of Things, vol. 28, Art. no. 101351, 2024, doi: 10.1016/j.iot.2024.101351.

[17] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," IEEE Access, vol. 8, pp. 106576–106584, 2020, doi: 10.1109/ACCESS.2020.3000421.

[18] A. Langley, M. Hamburg, and S. Turner, "Elliptic curves for security," RFC 7748, 2016, doi: 10.17487/RFC7748.

[19] H. Krawczyk and P. Eronen, "HMAC-based extract-and-expand key derivation function (HKDF)," RFC 5869, 2010, doi: 10.17487/RFC5869.

[20] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," RFC 2104, 1997, doi: 10.17487/RFC2104.

[21] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Advances in Cryptology—CRYPTO '96, LNCS 1109, pp. 1–15, 1996, doi: 10.1007/3-540-68697-5_1.

[22] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF protocols," RFC 8439, 2018, doi: 10.17487/RFC8439.

[23] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979, doi: 10.1145/359168.359176.

[24] Y. Liu et al., "Aion: Robust and efficient multi-round single-mask secure aggregation against malicious participants," in Proc. 34th USENIX Security Symposium, pp. 3025–3044, 2025.

[25] Z. Jiang, P. Ye, S. He, W. Wang, R. Chen, and B. Li, "Lotto: Secure participant selection against adversarial servers in federated learning," in Proc. 33rd USENIX Security Symposium, pp. 343–360, 2024.

[26] W. Yao, T. Zhou, Y. Han, and X. Wang, "Verifiable secure aggregation scheme for privacy protection in federated learning networks," Discover Computing, vol. 28, Art. no. 175, 2025, doi: 10.1007/s10791-025-09676-1.

[27] D. D. Truong, B. H. Pham, D. T. Ta, T. N. Phan, and T. A. T. Nguyen, "RADAR: Robust aggregation with dynamic adversarial-resilient routing decentralized federated learning," International Journal of Information Security, vol. 25, no. 3, Art. no. 86, 2026, doi: 10.1007/s10207-026-01257-7.

[28] E. T. Martínez Beltrán et al., "Mitigating communications threats in decentralized federated learning through moving target defense," Wireless Networks, vol. 30, no. 9, pp. 7407–7421, 2024, doi: 10.1007/s11276-024-03667-8.

[29] T. Li et al., "Federated optimization in heterogeneous networks," in Proc. MLSys, vol. 2, pp. 429–450, 2020.

[30] J. Wang et al., "Tackling the objective inconsistency problem in heterogeneous federated optimization," in Proc. NeurIPS, vol. 33, pp. 7611–7623, 2020.

[31] S. J. Reddi et al., "Adaptive federated optimization," in Proc. ICLR, 2021.

[32] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in Proc. ICML, PMLR, vol. 119, pp. 5132–5143, 2020.

[33] L. Leconte, M. Jonckheere, S. Samsonov, and E. Moulines, "Queuing dynamics of asynchronous federated learning," in Proc. AISTATS, PMLR, vol. 238, pp. 1711–1719, 2024.

[34] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," IEEE Access, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[35] Y. Meidan et al., "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.

[36] J. Wang, K. Yang, and M. Li, "NIDS-FGPA: A federated learning network intrusion detection algorithm based on secure aggregation of

gradient similarity models," PLOS ONE, vol. 19, no. 10, e0308639, 2024, doi: 10.1371/journal.pone.0308639.

[37] L. Cui and X. Wu, "ALDP-FL for adaptive local differential privacy in federated learning," Scientific Reports, vol. 15, Art. no. 26679, 2025, doi: 10.1038/s41598-025-12575-6.

[38] V. Kolluru et al., "Novel federated learning-based approach for network attack detection," Journal of Cloud Computing, vol. 15, Art. no. 62, 2026, doi: 10.1186/s13677-026-00882-w.

[39] T. Alsolami and M. Ilyas, "FedSMOTE-DP: Privacy-aware federated ensemble learning for intrusion detection in IoMT networks," Sensors, vol. 26, no. 5, Art. no. 1592, 2026, doi: 10.3390/s26051592.

[40] M. A. Khan, O. Khalid, and R. N. B. Rais, "Fed-DTCN: A federated disentangled learning framework for unsupervised zero-day anomaly detection in IoT with semantic-aware augmentation," Sensors, vol. 26, no. 6, Art. no. 1918, 2026, doi: 10.3390/s26061918.

[41] M. O. Nassar, "A robust and energy-efficient federated IDS for IIoT using spiking neural networks and differential evolution with adversarial resilience," Iraqi Journal for Computer Science and Mathematics, vol. 6, no. 4, Art. no. 3, 2025, doi: 10.52866/2788-7421.1328.

[42] M. O. Nassar and F. F. Al-Mashagba, "FED-SCADA: A trustworthy and energy-efficient federated IDS for smart grid edge gateways using SNNs and differential evolution," International Journal of Computer Network and Information Security, vol. 17, no. 6, pp. 1–15, 2025, doi: 10.5815/ijcnis.2025.06.01.

[43] M. I. M. Tanvir et al., "Privacy-preserving multimodal federated learning pipeline for cyber-resilient healthcare systems," PLOS ONE, vol. 21, no. 4, e0343669, 2026, doi: 10.1371/journal.pone.0343669.

[44] S. Alamro, "AnomLocal: A hybrid local-global anomaly detection model for network security using federated learning," PLOS ONE, vol. 21, no. 2, e0339981, 2026, doi: 10.1371/journal.pone.0339981.

[45] A. Komal and S. Li, "Intrusion detection in the Internet of Things: A comprehensive review of techniques, architectures, datasets, and emerging trends," Sensors, vol. 26, no. 11, Art. no. 3405, 2026, doi: 10.3390/s26113405.

[46] A. Allen, A. Mylonas, S. Vidalis, and N. Pitropakis, "Peer-to-peer federated learning: A comprehensive survey," AI, vol. 7, no. 7, Art. no. 243, 2026, doi: 10.3390/ai7070243.

[47] A. L. M. Serrano et al., "FedIHRAS: A privacy-preserving federated learning framework for multi-institutional collaborative radiological analysis with integrated explainability and automated clinical reporting," Biomedicines, vol. 14, no. 3, Art. no. 713, 2026, doi: 10.3390/biomedicines14030713.

APPENDIX A. ELABORATED FORMAL SECURITY ARGUMENTS

A. 1 Temporal-binding Game

The challenger provides authentication keys and answers commitment queries on arbitrarily chosen messages. The adversary wins by outputting a verifying tuple whose authenticated field vector has not been previously queried, or by producing a modified ciphertext whose authenticated digest has already been seen. The reduction keeps track of all queried field vectors and outputs a fresh forgery to the HMAC challenger when the adversary succeeds. A modified ciphertext with the same digest is also a collision for H.

A. 2 Replay and Slot Substitution

Round nonces are sampled independently and stored for reference alongside the replay cache. A replayed tuple from round t′ will either retain the original nonce and be rejected during the fresh round's verification step or contain a different nonce, round, or slot and be required to produce a valid tag. Slot substitution is similar, but the substituted value is s_(i,t).

A. 3 Secure Aggregation under Timing Rejection

Timing rejection allows the gateway to modify the dropout set before performing recovery. The gateway reveals G_t and M_t, and surviving clients release only the shares necessary to cancel the masks corresponding to M_t. The pairwise masks between accepted clients cancel out when summed, while the private masks of the accepted clients remain unreconstructed. The confidentiality of client contributions thus maintains the secure-aggregation confidentiality guarantee under the same collusion bound.

A. 4 Release-trace Game

The challenger samples a bit b hidden from the adversary, which determines whether the release-trace envelope is constructed around a real or dummy payload. In either case, the trace contains the same scheduled release time, size, authentication parameters, and encryption algorithm. Any advantage the adversary gains from distinguishing real from dummy traces must come from differences in timing, lengths, abort behaviors, or other implementation details. An empirical measurement of this advantage is given by the observer attack of Section VII-D.

Adaptive Virtual Reality Exposure Therapy Using AI-Driven Anxiety Prediction from Physiological and Behavioral Data

Fayaz Hussain G[1], Dr. P. Chenna Reddy[2]
Research Scholar, JNTUA, Anantapur, Assistant Professor, Dept. of CSE, Ravindra College of Engineering for Women, India[1]
Professor, Dept. of CSE, JNTUA College of Engineering, Ananthapuramu, Anantapur, India[2]

***Abstract*—All mental health disorders; anxiety disorders rank as the most common ones. They affect the cognitive functions of an individual as well as their ability to socialize or live in a quality manner. Exposure Therapy happens to be one of the most effective methods aimed at treating fears, phobias, and traumatic memories; however, the traditional approach to exposure therapy and VRET is unable to modify the exposure intensity in accordance with the situation that is occurring with the patient, thus making this method ineffective. This article suggests a solution to the problem by creating AVRET, or Adaptive Virtual Reality Therapy, which will provide the opportunity to predict the anxiety of the patient in real-time through their physiological (HR, HRV, GSR) and behavioral (eye-tracking, blink rate, pupil dilation) indicators. The suggested system uses an LSTM deep learning model to evaluate the amount of anxiety and change the scenario, distance to objects, sounds, or level of difficulties based on the current anxiety evaluation.**

***Keywords*—Virtual reality exposure therapy; LSTM; anxiety prediction; physiological computing; closed-loop systems; multimodal fusion**

I. Introduction

A cause of concern in the healthcare system is anxiety disorders that affect emotional well-being, cognitive abilities, and interpersonal relationships. Standard treatments for anxiety disorders include cognitive behavioral therapy (CBT) and exposure therapy; however, traditional exposure therapy does not have complete control over the environment, relies on the therapist's thorough assessment of the patient's condition, and does not provide tailored service, in addition to creating safety problems [1]. Virtual reality exposure therapy (VRET) overcomes some of the problems identified in traditional therapy by providing a safe environment for patients to practice the skills they acquired in therapy. Although VR exposure therapy has proved treatment effective, many of the existing systems remain rigid and unable to respond to the patient's physiological signals to adapt to the effectiveness of treatment. Advances in AI, deep learning, and physiological computing now allow emotional state to be inferred from signals such as heart rate (HR), heart rate variability (HRV), galvanic skin response (GSR), pupil diameter, eye tracking, eye blink rate, and motion.

This study proposes a closed-loop Adaptive VR Exposure Therapy (AVRET) framework that combines multimodal physiological sensing, behavioral analysis, LSTM-based anxiety prediction, and adaptive VR control with real-time feedback. An LSTM network extracts temporal patterns from the multimodal stream to estimate anxiety, and the adaptive controller uses this estimate to continuously tune scene complexity, object motion, loudness, and exposure level, improving personalization, comfort, safety, and interactivity relative to static VR therapy [2].

This research proposes a closed-loop intelligent healthcare framework combining physiological monitoring, behavioral analysis, deep-learning-based emotion assessment, adaptive VR control, and feedback, aimed at maximizing patient emotional comfort during therapy [3].

The main contributions of this work are: 1) A multimodal physiological and behavioral sensing approach for real-time anxiety measurement; 2) Integration of HR, HRV, GSR, eye-tracking, blink frequency, pupil dilation, and motion for holistic emotional awareness; 3) An LSTM-based deep learning model for temporal anxiety prediction; 4) A closed-loop VR controller that modifies the environment based on emotional state; 5) Improved personalization, emotional safety, and adaptive interaction in immersive healthcare; and 6) An AI-powered intelligent mental healthcare framework combining VR and deep learning.

II. Literature Survey

A. Prevalence and Exposure Therapy

Anxiety disorders affect an estimated 301 million people worldwide (WHO, 2023), including GAD, panic disorder, social anxiety disorder, specific phobias, agoraphobia, and PTSD, and account for 28.6 million DALYs annually [4]. In the United States alone, direct and indirect costs exceed $42 billion per year, yet fewer than 40% of patients receive effective treatment due to shortages of specialists, cost, stigma, and limited access [5].

Exposure therapy is still the standard treatment according to the reciprocal inhibition theory of Wolpe as well as modern models of inhibitory learning. The key factor behind the effectiveness of this technique is the intensity of the exposure, its performance without any attempts to escape the fear [6], and the level of predictability. However, application of the exposure method has a number of difficulties, such as real-world issues, lack of standardization, and thus reliance on subjective self-report tools, including the SUDS scale [7].

B. VRET and Physiological Affect Recognition

Virtual Reality Exposure Therapy (VRET), which was first created in the 1990s, allows the successful reproduction of real-life situations that provoke fear in a simulated 3-dimensional environment utilizing head-mounted displays and advanced techniques for tracking movements and spatial sounds, granting the doctor absolute control over the complexity and the intensity of the therapy process [8]. Reviews indicate the results of VRET are comparable to those obtained with in-vivo exposure, although it provides safety and accessibility for phobias, social anxiety, panic disorder, and PTSD. Nevertheless, the majority of existing systems are still based on rigid therapist-controlled procedures and have low adaptability towards rapid changes in users' emotional conditions.

It has been suggested that physiological computing can bridge the gap in anxiety and stress by activating the autonomic nervous system, bringing about measurable changes such as a rise in heart rate and a drop in heart rate variability, an increase in GSR (from normal skin conductance level and phasic responses), and changes in pupillary dilation, rate of blinking, and gaze pattern. Since no stand-alone solution has managed to capture the anxious state completely — cardiac signals are disturbed by motion; electrodermal measures have a slow response (and ocular signal changes are associated with attention) — using a combination of physiological signals can greatly direct the quest for an effective solution [9].

III. Problem Statement

Increasing stress, trauma, and lifestyle pressures have increased the rate of anxiety disorders, yet CBT and exposure therapy remain limited by dependence on therapists, poor scalability, and an inability to track patients' emotional state during treatment. VRET increases accessibility and control and has been used efficiently to treat phobias, PTSD, and social anxiety [10], but existing systems do not configure exposure to the patient's steadily changing emotional condition. Because emotional responses are highly individualized, static exposure risks either overexposure, increasing distress, or underexposure — limiting therapeutic effect. Furthermore, most anxiety-prediction approaches rely on a single physiological signal or offline analysis, limiting robustness, and common machine learning models struggle to capture long-term dependencies in physiological and behavioral data flow. These gaps motivate an Adaptive VR Exposure Therapy system that can track multiple modalities, predict anxiety in real time using deep learning, and adapt the environment continuously.

IV. Proposed Model

This study describes AVRET, a smart system for measuring a user's mood and making modifications to the VR environment based on the predicted level of anxiety. This is unlike traditional VR therapy, which uses preset conditions and relies on therapists, as AVRET automatically assesses physiological and behavioral parameters and adapts the environment in real-time using artificial intelligence [11]. Fig. 1 depicts a relevant closed-loop BCI VR setup whereby EEG is captured, feature extraction is done, and the classified state is returned to the user as visual, auditory, and haptic feedback. AVRET operates on the same closed-loop concept of functioning but relies on peripheral physiological signals as opposed to central signals.

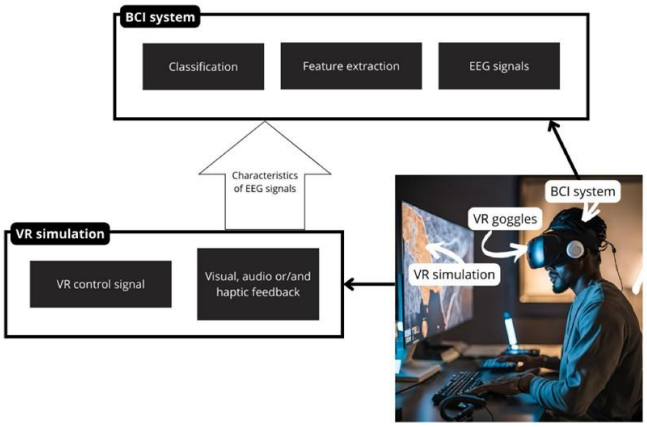


Fig. 1. Adaptive VR environment.

Fig. 2 shows the overall AVRET pipeline: User in VR Environment, Multimodal Sensor Input, Data Pre-processing, Feature Extraction, Multimodal Feature Fusion, LSTM-based Anxiety Prediction, Anxiety Classification, VR Adaptive Controller, VR Environment Adjustment, and a Feedback Loop.

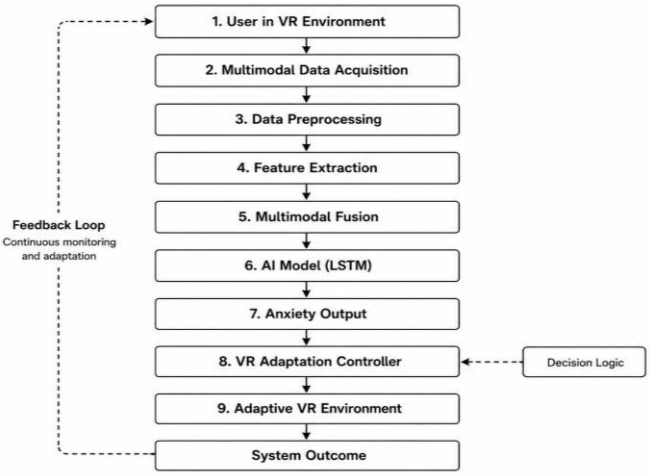


Fig. 2. Proposed adaptive virtual reality anxiety detection and adaptation pipeline.

The user is part of a virtual reality simulation (public speaking, crowds, heights, closed spaces, etc.) using head-mounted displays and motion tracking that enables emotional monitoring at any time. Multi-modal sensors continuously measure HR, HRV, GSR, eye-tracking, blink rate, pupil dilation, and head movement since anxiety, stress, and cognitive load occur simultaneously in multiple physiological systems. However, in order to use the raw signals, a number of processing techniques are applied to achieve proper filtering, smoothing, and alignment of the signals [12].

By extracting features, the latter signals can be converted into different statistical, temporal, and behavioral manifestations, such as mean/peak HR, HRV statistics, GSR peak attributes, blink frequency, pupil dilation change, fixation

time, and motion intensity. This way, cardiovascular, electrodermal, oculometric, and motion features are combined into a single emotional model using a fusion layer to make it more effective than one of the techniques [13]. The agreed features are then processed by the LSTM network that uses internal memory and does not analyze feature vectors like other models using the data multiplicity principle.

The level of anxiety predicted is sorted into Low, Medium and High and forwarded to the VR Adaptation Controller. In accordance with the level of anxiety, the VR adaptation controller alters the virtual environment accordingly: low level of anxiety – increases exposure intensity, medium level of anxiety – keeps the environment the same, high level of anxiety – lowers the stress level in the environment: scene complexity, lighting, object speed, sound, crowd density, stimulus object distance and exposure time are modified to reflect this adaptation. The process of continual monitoring leads to an adaptive approach which ensures the idea of personalization and the principle of comfort and safety of users. Constant feedback between monitoring, predicting, adapting, and re-evaluating proves that the system is more flexible as compared to static one-time VR therapy systems and methodologies [14]

V. Methodology

The AVRET framework proposed here is made of six stages as follows: obtaining the multimodal data, preprocessing the data, extracting features, combining the different modalities' features, performing anxiety prediction based on LSTM, and implementing a closed-loop adaptive VR control.

A. Data Acquisition, Preprocessing, and Key Formulas

In the process of virtual reality (VR) sessions, several multimodal physiological and behavioral signals are simultaneously collected in real time; these include the parameters of heart rate (HR), heart rate variability (HRV), reticular activation system (RAS), eye parameters (eye tracking), blink frequency, and pupil expansion. The captured data is filtered to remove noise, smoothed, excessive outliers removed, missing data filled in, and synchronized in time, ensuring a consistent format for analysis. Essential equations used for this purpose are as follows:

$$HR = 60 / RR \quad (RR = \text{interval between consecutive heartbeats})$$

$$HRV = \sqrt{[(1/N)\,\Sigma\,(RR_i - \text{mean}(RR))^2]} \quad \text{over } N \text{ samples}$$

$$GSR = SCL + SCR \quad (\text{tonic level} + \text{phasic response})$$

$$\text{Blink Rate} = \text{Number of Blinks} / \text{Time}$$

$$\text{Pupil Dilation (PD)} = P_{current} - P_{baseline}$$

$$\text{Normalization: } X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$$

B. Feature Extraction and Fusion

Cardiovascular features consist of average/peak HR, RR-interval and HRV statistics, and frequency domain parameters; electrodermal features include skin-conductance level, GSR peak frequency, amplitude, and recovery; behavioral/ocular features include eye blink rate, pupil-dilation rate, fixation duration, gaze pattern, and head-movement intensity. These are fused into a single vector, F_fusion = [F_HR, F_GSR, F_Eye], improving emotional encoding, accuracy and robustness over any single modality.

C. LSTM Anxiety Prediction

An LSTM network predicts anxiety from the sequential multimodal features because it retains sequential dependencies via internal memory cells rather than treating each feature vector independently [16]. Its gating equations are:

$$f_t = \sigma(W_f[h_{t-1},x_t]+b_f) \quad (forget)$$

$$i_t = \sigma(W_i[h_{t-1},x_t]+b_i) \quad (input)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1},x_t]+b_c) \quad (candidate)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (cell\ update)$$

$$o_t = \sigma(W_o[h_{t-1},x_t]+b_o) \quad (output)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (hidden\ state)$$

The model is trained with cross-entropy loss, Loss = $-\Sigma\, y_i \log(\hat{y}_i)$, using the Adam optimiser with backpropagation-through-time and early stopping. The final anxiety-class probability is A = Softmax(W_h h_t + b_h), related to Low, Medium, or High anxiety using thresholds T1 and T2 (Low: A<T1; Medium: T1≤A<T2; High: A≥T2). Increased heart rate (HR) and decreased heart rate variability (HRV) are significant indicators of excitement and anxiety [15], and the LSTM tracks the speed of the occurrences of emotional changes, transitions, and different time intervals relating to stress from rising to recovery to compute anxiety intensity during virtual reality immersion [16].

D. Adaptive VR Control and Feedback

According to the level of anxiety predicted, the VR Adaptation Controller alters the complexity of the scene, speed of movement of the objects, sound level, density of crowd, time of exposure, brightness, distance of stimulus, and the difficulty in performing the tasks. It decreases the stimulation of the environment in case anxiety is high and gradually increases it in case anxiety is low, which ensures that the therapy maintains a stable emotional background. The feedback loop (physiological measurement → behavioral analysis → anxiety level prediction → adaptation of the VR to the patient → reassessment of the emotions of the patient) provides for the continuous operation of the system throughout the treatment process, which is not typical for the traditional therapy based on static models [17].

VI. Experimental Setup

The evaluation of the experiment validated the AVRET framework for anxiety prediction accuracy, emotional responsiveness, adaptive VR effectiveness, deep learning performance, personalization capability, and system reliability.

The hardware involved a VR headset, cardiovascular and GSR sensors, an eye-tracking device, motion-tracking sensors, and a GPU-based workstation for the inference process.

The software used Python, TensorFlow/Keras for LSTM-based modeling, OpenCV for behavioral analysis,

NumPy/Pandas for signal processing, Matplotlib for visualization, and Unity/Unreal Engine for VR scene rendering.

A synchronized multimodal dataset consisting of HR, HRV, GSR, eye tracking, blink, pupil, and motion signals was acquired from VR sessions with different anxiety levels. The dataset was divided into training, validation, and testing datasets. Six VR scenarios were used where the stimulus was changed depending on the predicted level of anxiety.

Using the Adam optimizer, cross-entropy loss, backpropagation through time, cross-validation, and early stopping, the LSTM model was trained on multimodal feature windows in a particular order. Performance was measured through accuracy, precision, recall, F1-score, confusion-matrix analysis, and real-time response latencies. The adaptive VR controller was tested regarding its ability to alter the complexity of the environment, density of people in the hall, exposure duration, volume of sound, speed of object movement, and visual stimuli with regard to the level of anxiety compared to a commonly used (non-adaptive) VR therapy.

The experimental process involves user checking, turning on the sensors, going through collecting the data, processing the data, studying the features, using LSTM to provide its predictions of anxiety, and monitoring the situation.

VII. Results

The evaluation of the AVRET system was conducted with respect to the parameters of anxiety prediction accuracy, adaptability, emotional responsiveness, deep learning convergence, personalization, and reliability. The research results conducted with the use of multimodal sensing and deep learning show that the reliability and adaptability level achieved is much higher than that typical of static VR therapy.

A. Anxiety Prediction Accuracy

The LSTM model has generated good predictions for anxiety from the different intensity sessions in VR, utilizing temporal-sequence processing, multimodal integration, continuous context, and its capability to adapt to emotional patterns. Table I summarizes the classification performance.

TABLE I. Evaluation Metrics

Metric	Performance
Accuracy	90.0%
Precision	94.8%
Recall	94.1%
F1-Score	94.4%
Latency	0.8 sec

B. Training Performance

Training accuracy and loss trends (Fig. 3–4) suggest a regular pattern of convergence since accuracy improved steadily again and loss decreased – thus, the LSTM learned the patterns of stress development, fluctuation, escalation, and recovery processes without overfitting.

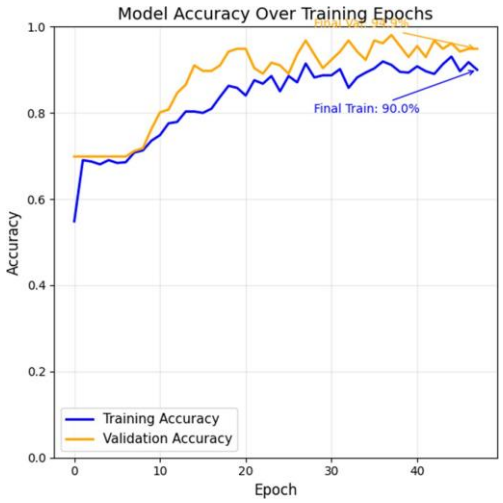


Fig. 3. Training accuracy curve of the proposed LSTM model.

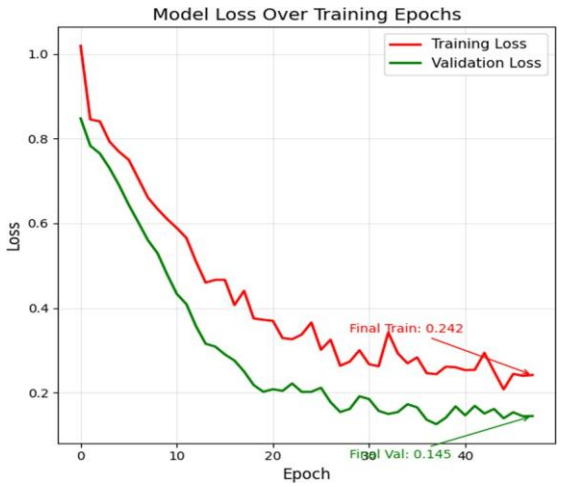


Fig. 4. Training loss curve of the proposed LSTM model.

C. Confusion Matrix Evaluation

Confusion-matrix analysis (Fig. 5–6) shows strong detection consistency with Low, Medium, and High anxiety classes with minimal misclassification, confirming the system's sensitivity to stress and emotional discomfort during VR immersion [13].

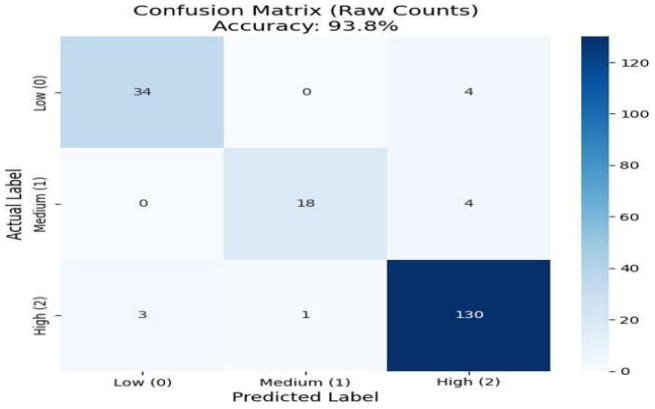


Fig. 5. Confusion matrix for anxiety classification.

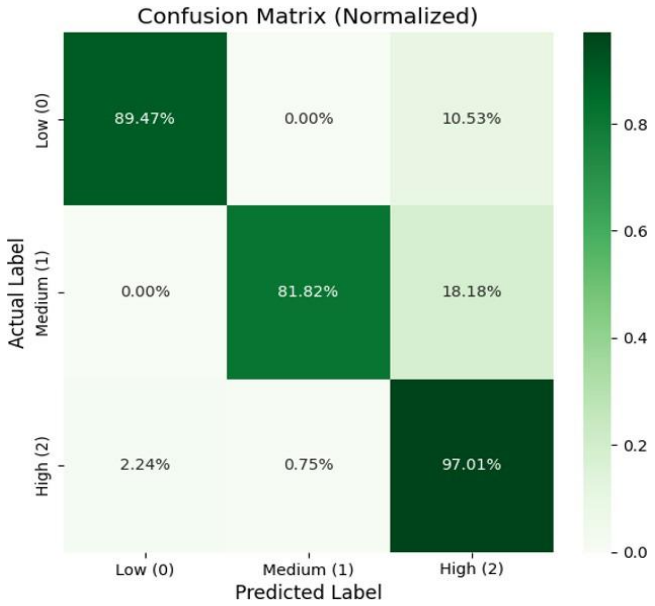


Fig. 6. Normalized confusion matrix for anxiety classification.

D. Multimodal Fusion Performance

Fusing HR, HRV, GSR, eye-tracking, and motion features substantially outperformed single-modality prediction (Table II), confirming that multimodal fusion enhances robustness and classification reliability.

TABLE II. COMPARISON OF PREDICTION APPROACHES

Method	Accuracy
Heart Rate Only	78.4%
GSR Only	80.1%
Eye Tracking Only	76.9%
Multimodal Fusion	95.2%

E. Adaptive VR and Physiological Signal Analysis

The adaptive controller successfully modulated scene complexity, exposure difficulty, sound intensity, object movement, crowd density, and environmental stimulation based on predicted anxiety [15]. Continuous multimodal monitoring (Fig. 7) shows HR rising, HRV declining, and GSR, pupil dilation, and eye blink rate varying under high-stress scenarios such as crowds, public speaking, and height exposure, showing that multimodal fusion achieves more reliable emotional representation than any single modality.

F. Adaptive VR Exposure Analysis

TABLE III. CLASSIFICATION REPORT FOR ANXIETY PREDICTION

Class	Precision	Recall	F1	Supp.
Low	91.9%	89.5%	90.7%	38
Medium	94.7%	81.8%	87.8%	22
High	94.2%	97.0%	95.6%	134

The controller continuously divided emotional state into Maintain, Reduce, or Increase Exposure based on the LSTM's predicted anxiety trajectory (Table III, Fig. 8), reducing response as anxiety increased and increasing it as the user's emotional state improved, ensuring real-time responses, dynamic personalization, continuous stress regulation, and improved emotional safety.

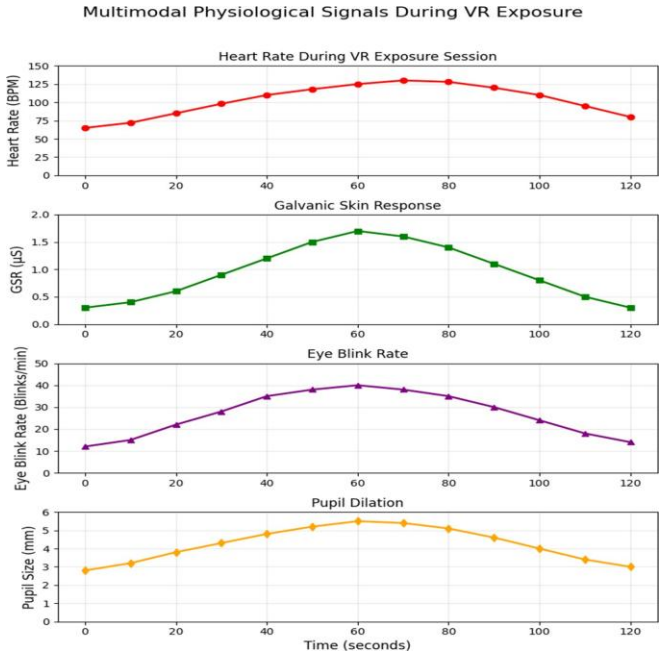


Fig. 7. Variation in multi-modal physiological signals during VR exposure.

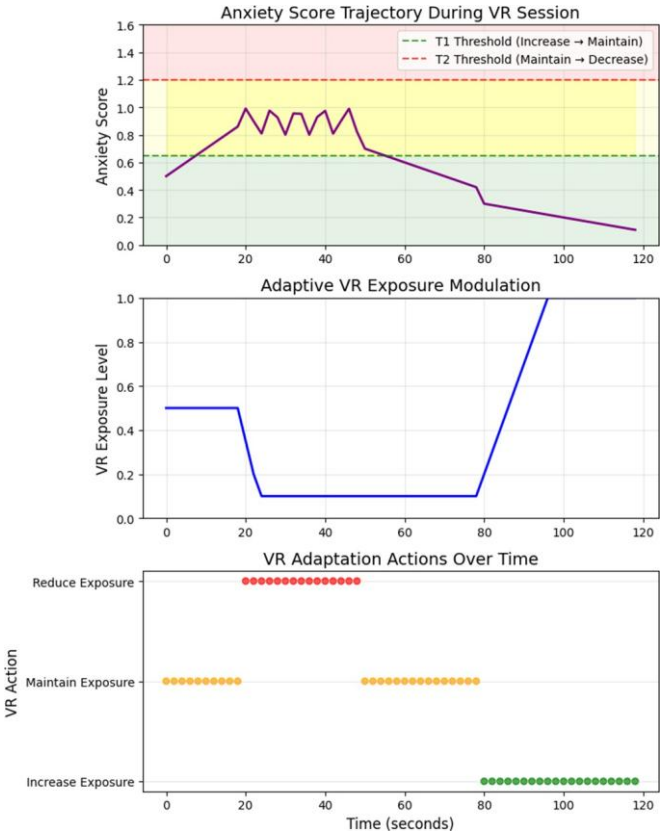


Fig. 8. Adaptive VR exposure modulation based on anxiety prediction.

G. Comparative Analysis

Table IV compares AVRET with existing approaches. Unlike traditional VRET (manual control, no adaptation) and single-modality predictors (HR-only, GSR-only, EEG-only, or eye-tracking-based models with partial or no adaptation), AVRET combines seven physiological/behavioral channels with an LSTM predictor and fully adaptive VR control, achieving markedly higher accuracy and confirming its advantage as an advanced, intelligent emotional-healthcare system.

TABLE IV. COMPARATIVE ANALYSIS

Method	Modalities	Model	RT Adapt.	VR Adapt.	Acc.
Traditional VRET	None	Manual	No	Static	Low
HR-Based	HR only	SVM	Partial	No	Moderate
GSR-Based	GSR only	RF	No	No	Moderate
EEG Emotion	EEG	CNN	Partial	No	High
Behavioral	Eye	RNN	Partial	Limited	Moderate
Proposed AVRET	Multimodal	LSTM	Yes	Full	Very High

VIII. DISCUSSION

These results support the effectiveness of multimodal physiological sensing, deep-learning-based anxiety prediction, and adaptive VR interaction for intelligent mental healthcare. This performance was driven by the LSTM architecture, which effectively learns temporal relationships among observations; unlike models that analyze each observation independently, it captured temporal dependencies in the design and resolution of anxiety, improving real-time prediction and adaptation. The adaptive VR controller used these predictions to continuously adjust exposure intensity, modifying environmental complexity, sound, and exposure duration to keep interactions emotionally balanced and avoid both overexposure and underexposure.

IX. CONCLUSION

This study proposed AVRET, a closed-loop intelligent healthcare system combining multimodal physiological sensing, behavioral analysis, LSTM-based deep learning, and adaptive VR control. The system collects HR, HRV, GSR, eye-tracking, blinking rate, pupil dilation, and other data continuously during VR therapy, processes it through a series of preprocessing, feature fusion, and deep-learning steps, and makes real-time predictions of anxiety. The experimental results demonstrate that AVRET is superior to static VR therapy in terms of prediction accuracy, emotional responsiveness, and adaptive personalization, with adaptive controller able to adjust exposure intensity based on changes in emotional state.

X. FUTURE SCOPE

This framework provides a foundation for future work to validate PA-VRET (EEG, EMG, facial expression) for richer emotional and cognitive-state recognition, and reinforcement learning or generative AI to automatically generate VR exposure environments based on patient progress. Future directions include extending AVRET to post-traumatic stress disorder, depression, workplace stress, pediatric mental healthcare, and telehealth applications using emotionally responsive VR.

REFERENCES

[1] Y. L. Tan et al., "Virtual reality exposure therapy for social anxiety disorders: a meta-analysis and meta-regression of randomized controlled trials," Anxiety, Stress & Coping, vol. 38, no. 2, pp. 141–160, 2025, doi: 10.1080/10615806.2024.2392195.

[2] W. Zeng, J. Xu, J. Yu, and X. Chu, "Effectiveness of virtual reality therapy in the treatment of anxiety disorders in adolescents and adults: a systematic review and meta-analysis of randomized controlled trials," Frontiers in Psychiatry, vol. 16, p. 1553290, 2025, doi: 10.3389/fpsyt.2025.1553290.

[3] E. Carl et al., "Virtual reality exposure therapy for anxiety and related disorders: A meta-analysis of randomized controlled trials," J. Anxiety Disorders, vol. 61, pp. 27–36, 2019, doi: 10.1016/j.janxdis.2018.08.003. This is the real update to Powers and Emmelkamp, so it can stand in for your [1]. The original is Powers and Emmelkamp, J. Anxiety Disorders, vol. 22, no. 3, pp. 561–569, 2008.

[4] J. Sameri, H. Coenegracht, S. Van Damme, F. De Turck, and M. Torres Vega, "Physiology-driven cybersickness detection in virtual reality: a machine learning and explainable AI approach," Virtual Reality, vol. 28, art. 174, 2024, doi: 10.1007/s10055-024-01067-z.

[5] A. Baldini, E. Patron, C. Gentili, E. P. Scilingo, and A. Greco, "Novel VR-based biofeedback systems: a comparison between heart rate variability- and electrodermal activity-driven approaches," IEEE Trans. Affective Computing, 2024.

[6] F. Li and D. Zhang, "Transformer-Driven Affective State Recognition from Wearable Physiological Data in Everyday Contexts," Sensors, vol. 25, no. 3, 2025, doi: 10.3390/s25030761.

[7] Premkumar P, Heym N, Myers JAC, Formby P, Battersby S, Sumich AL, Brown DJ. Augmenting self-guided virtual-reality exposure therapy for social anxiety with biofeedback: a randomised controlled trial. Front Psychiatry. 2024 Nov 12;15:1467141. doi: 10.3389/fpsyt.2024.1467141. Erratum in: Front Psychiatry. 2025 Feb 25;16:1548762. doi: 10.3389/fpsyt.2025.1548762. PMID: 39600795; PMCID: PMC11589380.

[8] M. K. Moser, M. Ehrhart, and B. Resch, "An Explainable Deep Learning Approach for Stress Detection in Wearable Sensor Measurements," Sensors, vol. 24, no. 16, 2024, doi: 10.3390/s24165085

[9] X. Li and M. Mahmoud, "Interpretable Concept-based Deep Learning Framework for Multimodal Human Behavior Modeling," arXiv:2502.10145. It is a concept-based framework for interpretable affective computing, and it is a preprint.

[10] J. Y. Chun et al., "Prediction of Specific Anxiety Symptoms and Virtual Reality Sickness Using In Situ Autonomic Physiological Signals During Virtual Reality Treatment in Patients With Social Anxiety Disorder: Mixed Methods Study," JMIR Serious Games, vol. 10, no. 3, e38284, 2022, doi: 10.2196/38284

[11] F. Wilhelm, M. Pfaltz, J. Gross, I. Mauss, S. Kim, and B. Wiederhold, "Mechanisms of Virtual Reality Exposure Therapy: The Role of the Behavioral Activation and Behavioral Inhibition Systems," Applied Psychophysiology and Biofeedback, vol. 30, no. 3, pp. 271–284, 2005.

[12] G. Fayaz Hussain and P. Chenna Reddy, "Exposure Therapy in the Treatment of Phobias Using Virtual Reality," Proceedings of the Third International Conference on Cognitive and Intelligent Computing, Volume 2, Cognitive Science and Technology, Springer, Singapore, 2025, Available: https://doi.org/10.1007/978-981-97-9266-5 23

[13] H. R. Saeidnia, S. G. Hashemi Fotami, B. Lund, and N. Ghiasi, "Ethical Considerations in Artificial Intelligence Interventions for Mental Health and Well-Being: Ensuring Responsible Implementation and Impact," Social Sciences, vol. 13, no. 7, p. 381, 2024, doi: 10.3390/socsci13070381.

[14] Kolding, S., Lundin, R. M., Hansen, L., & Østergaard, S. D. (2025). Use of generative artificial intelligence (AI) in psychiatry and mental health

care: a systematic review. Acta Neuropsychiatrica, 37, e37. doi:10.1017/neu.2024.50.

[15] K.-Y. Chang and C.-F. Lee, "Enhancing Virtual Restorative Environment with Generative AI: Personalized Immersive Stress-Relief Experiences," Lecture Notes in Computer Science, vol. 15791, pp. 132–144, 2025, doi: 10.1007/978-3-031-93502-2_9.

[16] G. F. Hussain and A. Heena, "Exploring the Impact of Physical vs. Artificial Motion on Immersion, Usability, and Cybersickness in Virtual Environments," 2024 2nd International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), Hyderabad, India, 2024, pp. 1-6, doi: 10.1109/ICAITPR63242.2024.10960045.

[17] Biomarkers in exposure-based treatment of anxiety in virtual reality: a systematic review, Frontiers in Virtual Reality, doi: 10.3389/frvir.2024.1355082.

Predictive RFID Tour Monitoring Dashboard Integrating Multimodal Machine Learning and Contextual Data

Heng Jin Koey[1], Siti Zuraidah Ibrahim[2]*, Ismahayati Adam[3],
Megat Syahirul Amin Megat Ali[4], Rashidah Che Yob[5], Sugchai Tantiviwat[6], Sazwan Syafiq Mazlan[7]
Faculty of Intelligent Computing (FKC), Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia[1, 2]
Faculty of Electronic Engineering and Technology (FKTEN), Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia[3, 5]
Microwave Research Institute, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia[4]
Faculty of Industrial Education and Technology, Rajamangala University of Technology Srivijaya (RUTS), Songkhla, Thailand[6]
Faculty of Engineering, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia[7]

***Abstract*—Traditional tourism monitoring systems rely on manual tracking and overlook dynamic factors influencing travel duration. To address this limitation, this research presents an intelligent web dashboard integrating RFID infrastructure and multimodal machine learning to assess the risk of delay in tourist activities. The project encompasses the design and physical prototyping of a Radio Frequency Identification (RFID) based checkpoint system. This prototype demonstrated functional detection of members upon entry and exit, providing an automated identification layer. To predict delays, the system processes two diverse datasets. A biometric gait dataset facilitates human speed prediction utilizing age, gender, body mass, body size, and shoe size. Simultaneously, a vehicle dataset provides environmental context through travel distance, weather ratings, traffic conditions, and vehicle classification. The study evaluates Ridge Regression, Random Forest, and Gradient Boosting Regressor algorithms for human speed prediction, while Random Forest, Gradient Boosting Regressor, and Long Short-Term Memory (LSTM) networks are utilized for vehicle speed prediction. Evaluation revealed that the Gradient Boosting Regressor achieved the lowest error metrics across both datasets. This superior model was exported and implemented into a Python Flask backend to generate live transit time estimations. The dashboard provides live visibility into member locations and implements automated missing member notifications. This allows tour organizers to respond promptly to movement irregularities without manual tracking. Overall, the results demonstrated the feasibility of the proposed automated tour group monitoring system. However, the current prototype was validated through controlled functional testing, and further multi-user and field-based evaluation is required to establish its scalability and generalizability for larger smart tourism environments.**

Keywords*—RFID; predictive analytics; multimodal machine learning; contextual data; smart tourism

I. Introduction

Tour group management often requires continuous coordination of participants during movement between destinations, especially when groups are required to gather at specific checkpoints before continuing to the next activity. Traditional approaches using manual roll calls, name lists, and verbal confirmations need direct human intervention and may result in inaccurate or delayed attendance information. Such methods are time-consuming and susceptible to human error, particularly when attendance must be repeatedly verified throughout a journey [8]. These limitations become more significant as the number of participants and checkpoints increases, since it becomes difficult to accurately verify the presence of every individual when the group moves between different locations. The absence of an automated tracking mechanism may result in members being unintentionally left behind, leading to delays, confusion, dissatisfaction, and potential safety issues [11].

Digital communication tools and mobile messaging applications can provide an alternative method for sharing attendance updates; however, these approaches still depend heavily on manual input and coordination. Group leaders may need to request individual responses from participants to confirm their arrival or departure, while participants may forget to respond, provide inaccurate information, or experience connectivity problems. Such methods also provide limited automated verification and structured historical records for confirming whether a participant has passed through a specific checkpoint. Consequently, when a member arrives late or fails to reach a designated location, group leaders may have difficulty determining whether the member is still travelling, delayed, or missing. This creates a conflict between maintaining the planned travel schedule and ensuring that all participants remain accounted for throughout the journey.

Radio Frequency Identification (RFID) provides an automated identification mechanism by assigning a unique identifier to each participant. RFID-based systems can perform automatic data capture and reduce the need for manual attendance verification. Instead of calling names or checking participant lists individually, members can present their RFID tags at designated checkpoints, allowing their attendance and checkpoint information to be recorded electronically. RFID can therefore provide structured identification records that can subsequently be transmitted to a centralized monitoring platform. Previous work has demonstrated the use of RFID with embedded hardware and cloud-based data logging to automatically record participant attendance, while RFID-based

*Corresponding author.

tracking systems can also be extended to provide centralized monitoring of member movements.

However, RFID identification alone does not provide sufficient support for tour management. A conventional RFID checkpoint system records whether a participant has been detected at a particular location but does not directly determine whether the participant is likely to arrive at the next checkpoint on time. Furthermore, when several checkpoints are operating simultaneously, the collected information needs to be synchronized and presented in a form that can be easily interpreted by tour leaders. Data synchronization and communication latency are therefore important considerations in real-time tracking systems, particularly when multiple devices are required to transmit information to a centralized platform [11]. A monitoring system must therefore not only collect RFID data but also provide timely and reliable updates to support decision-making during a journey.

Collected checkpoint records generally represent historical or status information and may require additional analysis to determine potential delays. A participant's latest recorded checkpoint alone does not indicate how long the participant will need to reach the next destination. Machine learning can complement RFID-based monitoring by learning relationships between participant characteristics, transportation conditions, and travel speed. The predicted travel performance can subsequently be used to estimate travel time and compare the expected arrival time with the checkpoint deadline. This provides predictive information that can assist group leaders in identifying participants who may be at risk of arriving late rather than relying solely on a missing-member status after the deadline has passed.

Therefore, this study proposes an RFID-Based Tour Group Tracking and Predictive Monitoring System that integrates RFID checkpoint identification, a cloud-based database, real-time web monitoring, automated missing-member notification and machine learning-based travel prediction. The RFID checkpoints record member identification at designated locations and transmit the records to a centralized database, while the dashboard synchronizes the information and presents the current member status to administrators and tour leaders in real time. The proposed architecture therefore combines physical identification, cloud data management, real-time communication, and predictive analysis within a single monitoring system.

II. Literature Review

Radio Frequency Identification (RFID) uses radio waves to identify and track tagged objects without physical contact or line-of-sight, a capability that traces back to World War II friend-or-foe radar identification [1]. A typical RFID system consists of a transponder (tag), a scanning antenna, and an emitter–receiver that together form the reader, or interrogator [2]. The reader broadcasts a radio-frequency signal to power and activate the tag, then demodulates the returned signal into digital data for downstream processing [3]. Passive high-frequency (HF) tags are preferred for short-range, contact-triggered check-in/check-out applications because they require no onboard battery, resist interference in crowded environments, and remain cost-effective for multi-checkpoint deployment, unlike ultra-high-frequency (UHF) or microwave tags who's longer read range would allow a member to pass a checkpoint undetected.

Research found that commercial group-management platforms illustrate current approaches to traveler tracking. VNFCPass assigns each traveler an NFC bracelet or badge, letting operators import participant lists, verify presence with a single tap, and manage multiple groups in parallel [4]. TourTrax combines GPS positioning with checkpoint scanning and two-way communication to provide real-time alerts and task-completion reporting for patrol-style tour operations [5]. While effective for group visibility, neither platform incorporates predictive analytics to flag members at risk of missing a scheduled deadline, a gap this study addresses. Several studies apply RFID and related IoT sensing to attendance and tracking problems. Farag and Abouelela [6] implemented a low-cost passive-RFID attendance system that logs check-in events with minimal hardware overhead. Sunder and Am [7] developed an RFID-based asset-management platform for tracking equipment across a facility. Gopinath et al. [8] proposed a real-time group-coordination system that alerts operators when a member deviates from the group. Belka et al. [9] designed a Bluetooth Low Energy (BLE) indoor-positioning system with overlap-resistant beacon placement for tourism navigation, reporting improved localisation accuracy over single-beacon layouts. Hutabarat [10] combined RFID wristbands for indoor detection with smartphone GPS for outdoor tracking and IP cameras for visual confirmation across a large recreational area, while Yadav [11] logged ESP32/RC522 RFID scans directly to Google Sheets for lightweight cloud attendance recording. Camera-based alternatives that use the MATLAB face-recognition system of Pamith et al. [12], the Flutter/Wi-Fi proximity system of Mahadik et al. [13], and the Flask–Supabase face-recognition system of Ibezim [14] achieve tag-free identification but remain sensitive to lighting, network reliability, and privacy concerns. Collectively, these works confirm RFID as a low-latency, low-power solution for controlled indoor and semi-outdoor checkpoint tracking, though few integrate a predictive layer for anticipating schedule risk.

Beyond detection, several works apply ML/DL to strengthen the predictive value and reliability of sensor data. Abbass et al. [15] used a Long Short-Term Memory (LSTM) network to predict power-system voltage stability from historical grid data, reporting high accuracy but also flagging overfitting risk on limited datasets, a consideration directly relevant to the temporal regression models developed in this study. Wang [16] proposed a CNN-based adaptive-modulation and error-correction scheme for chip-less RFID tags, showing that convolutional pattern recognition can raise signal-to-noise ratio and reduce bit-error rate compared with PSO- and 1D-CNN-based decoding, demonstrating that deep learning can improve the reliability of RFID-derived data even at the physical-signal level. In a related embedded-IoT context, Tomari et al. [17] trained YOLOv8/v9/v10 detectors to identify recyclable items on a CPU-only automated recycling machine, selecting YOLOv8-m as the best accuracy–efficiency trade-off for real-time inference on constrained hardware; the same accuracy-versus-resource trade-off suggests this project's

choice of lightweight regression models (Ridge Regression, Random Forest, Gradient Boosting) alongside LSTM for a resource-constrained Flask backend.

The reviewed literature confirms RFID as a mature, low-power solution for indoor/controlled-environment tracking [1]– [3]; the asset-management paper [4], [7] and RFID/IoT attendance systems [6]– [14] can reliably log checkpoint events. Separately, ML/DL techniques have improved prediction accuracy and signal reliability in adjacent IoT domains [15]– [17]. However, none of the reviewed group-tracking or RFID-attendance systems combine real-time checkpoint logging with a machine-learning layer that predicts individual travel risk, such as the likelihood of missing a schedule deadline, based on biometric, distance, and contextual features. This gap motivates the proposed RFID-based tour-tracking dashboard, which integrates checkpoint-level RFID logging with a two-stage ML pipeline for walking/vehicle speed and deadline-risk prediction.

III. METHODOLOGY

A. Project Design Flow

Each tour member is assigned a unique passive RFID tag. At each checkpoint, an RC522 RFID reader connected to an ESP32 detects the tag UID when the member taps the card to the reader. The ESP32 validates the received UID, filters repeated scans, associates the scan with the corresponding checkpoint, and records the timestamp. A valid scan is then transmitted through the ESP32's Wi-Fi connection to the Supabase backend in JSON format. The received RFID data is processed through the Supabase REST API and stored in the PostgreSQL database.

The React-based web dashboard retrieves and synchronizes these records using Supabase. Supabase Realtime enables new RFID events and status changes to be reflected on the dashboard without requiring manual page refreshes. The dashboard displays each member's status and checkpoint information, allowing administrators and group leaders to monitor the tour in real time. When a member fails to reach the designated checkpoint before the configured deadline, the system updates the member's status to “Missing” and generates an alert.

The system includes predictive analysis to estimate travel performance and potential delay. Machine learning models are trained using participant and transportation-related data to predict walking speed and vehicle speed. When a user requests a risk assessment, the React dashboard sends the required member and route information to the Flask backend through a RESTful API. The backend processes the input using the trained model and estimates the travel speed and corresponding travel time. The predicted travel time is compared with the predefined checkpoint deadline to determine the member's potential delay risk. The prediction results are then returned to the dashboard and presented through the Analytics Page.

B. System Architecture

The proposed system is an integrated hardware-software platform consisting of four main layers: embedded hardware, cloud database, frontend application and machine learning backend, as shown in Fig. 1. The overall architecture enables RFID data collected at physical checkpoints to be transmitted to the cloud database, synchronized with the web dashboard in real time, and used for travel-time and delay-risk prediction.

The embedded hardware layer collects RFID checkpoint events, while the cloud database layer stores and manages system data. The frontend application provides interfaces for member management, real-time monitoring, route configuration, and analytics. The machine learning backend processes prediction requests from the frontend and provides travel-time and delay-risk results. These layers work together to provide an automated system for monitoring tour group movements and identifying potential delays.

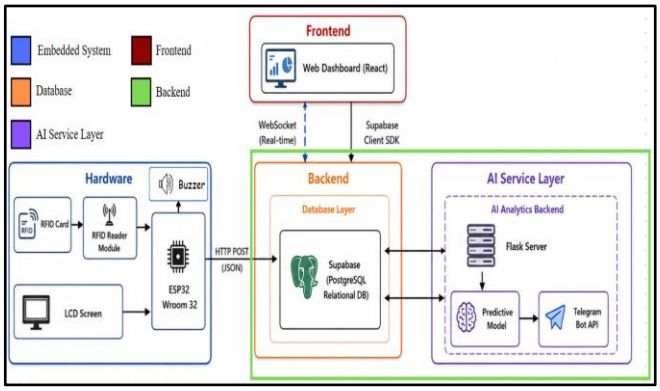


Fig. 1. Block diagram of the integrated RFID-based tracking and monitoring system for tour group members.

C. RFID Checkpoint Hardware Implementation

The RFID checkpoint was designed as a portable identification station that can be deployed at different locations during a group journey. The ESP32-WROOM-32 was selected as the central microcontroller because it provides integrated Wi-Fi connectivity and supports the communication interfaces required to connect the RFID reader and display. The RC522 reader was selected as the HF RFID reader, while passive RFID cards were used as member identification tags. The use of passive tags eliminates the requirement for individual tag batteries and is therefore suitable for repeated use by tour participants.

The RC522 communicates with the ESP32 through the SPI interface. The RFID reader is connected to the ESP32 through the corresponding power, reset, clock, master-in/slave-out, master-out/slave-in, and chip-select lines. An I²C LCD is connected separately to provide visual feedback, while the buzzer is connected to a GPIO output for audible confirmation. The hardware arrangement allows the ESP32 to simultaneously communicate with the RFID reader and peripheral feedback devices.

Fig. 2 shows the flowchart of mode operations at the checkpoint. Two operating modes are implemented at the RFID checkpoint: the registration mode and tracking mode. During registration, an administrator scans an RFID card and associates its UID with the selected participant profile. During tracking, the system verifies whether the scanned UID has been registered. For a valid UID, the system records the participant's identity, checkpoint, and timestamp and provides successful-

scan feedback through the LCD and buzzer. The RFID scan record is subsequently transmitted to the cloud database for further processing.

This approach allows the physical checkpoint to function independently as a data acquisition device while the database and web application handle storage, visualization, and higher-level analysis.

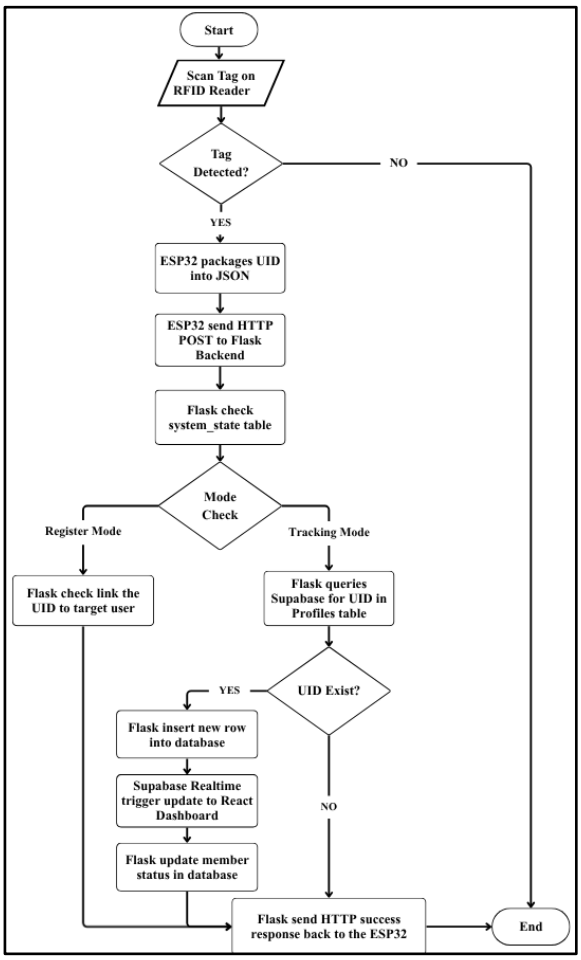


Fig. 2. Flowchart of track and register mode at checkpoint.

D. Cloud Database and Real-Time Monitoring

The Supabase PostgreSQL database serves as the central storage layer for the system. The database contains structured information covering participant profiles, RFID identifiers, travel groups, checkpoints, travel sessions, and system states. RFID scan records contain the information required to determine a participant's most recent confirmed checkpoint and movement history.

After an RFID scan is successfully processed by the ESP32, the scan record is inserted into the database through an HTTP request. The database update then becomes the source event for the real-time monitoring system. Instead of requiring the dashboard to repeatedly reload the database, Supabase Realtime is used to detect changes to subscribed tables and propagate them to connected clients.

The React dashboard maintains subscriptions to relevant database changes. When a new RFID record is inserted, or an existing participant status is updated, the corresponding dashboard components refresh their displayed information. This allows the current checkpoint, member status, and group attendance information to be presented without requiring manual page refreshes.

The dashboard also provides administrative functions for creating and modifying members, groups, checkpoints, and routes. Administrators can configure checkpoint locations, assign members to groups, and create a designated gathering or exit checkpoint. Group leaders can use the monitoring interface to observe participant status and identify members who have not reached the required checkpoint.

E. Real-Time Communication and Missing-Member Detection

Two communication mechanisms were considered for synchronizing checkpoint changes with the dashboard which are HTTP polling and WebSocket communication. In the polling approach, the client periodically sends HTTP requests to check whether new data is available. The system used a three-second polling interval during performance evaluation. Although polling is straightforward to implement, updates can remain undetected until the next polling cycle, and shorter polling intervals increase the number of requests and network traffic.

The proposed system therefore uses Supabase Realtime WebSocket communication for the deployed monitoring function. Supabase Realtime uses PostgreSQL logical replication to detect subscribed INSERT and UPDATE operations. When a relevant database record changes, the update is pushed to connected clients through a persistent WebSocket connection. This event-driven approach is suitable for RFID checkpoint monitoring because scan events occur irregularly and may require immediate notification.

A missing-member detection mechanism is implemented using the checkpoint deadline defined for a travel route. When the system determines that a participant has not reached the designated checkpoint within the required time, the participant status can be updated to “Missing”. The status is displayed on the dashboard, and a notification can be generated for the administrator. The notification provides information such as the participant's name, group and contact information to support follow-up action.

F. Route Management and Travel-Time Prediction

The system includes a route-management module that allows administrators or group leaders to define a sequence of checkpoints before a journey begins. Each route is divided into segments, with each segment containing a starting checkpoint, destination checkpoint, and transportation mode. The transportation modes supported include walking, vehicle, and bus.

After a route has been created, the user can select a participant and initiate a risk prediction. The system sends the participant's profile information, route information, checkpoint sequence, and transportation modes to the Flask prediction

backend. The backend calculates the expected travel duration for each route segment according to its assigned transportation mode. The predicted segment durations are then combined to obtain the estimated total journey time.

For a designated exit or gathering checkpoint, the estimated arrival time is compared with the configured deadline. The difference between the predicted arrival time and the required arrival time is used to determine whether the participant is likely to be late. The resulting prediction is returned to the React dashboard, where the estimated journey duration, remaining time, risk percentage, route status, and segment-level prediction are displayed.

This prediction process allows the system to provide information before and during a journey rather than relying solely on historical RFID records. When a new RFID scan is received during the journey, the participant's latest confirmed checkpoint can be used to update the prediction according to the remaining route. The following equations define the estimated travel time and the corresponding deadline-utilization ratio.

$$T_{estimated} = \frac{D}{V_{predicted}} \quad (1)$$

where:

- D = remaining travel distance
- $V_{predicted}$ = predicted walking or vehicle speed
- $T_{estimated}$ = estimated time required to reach the destination/checkpoint

and

$$R = \frac{Tneeded}{T_{remaining}} \times 100\ \% \quad (2)$$

where:

$T_{remaining}$= time remaining until the relevant deadline

R = risk-utilization ratio, expressed as a percentage

G. Machine Learning Model Development

Fig. 3 shows the pipeline of the machine learning process. The machine learning component was developed to estimate participant travel speed and support delay-risk assessment. The development process consisted of data collection, preprocessing, dataset splitting, model training, hyperparameter tuning, performance evaluation, model selection and deployment. The full pipeline follows the process of preparing the datasets, training multiple candidate models, evaluating their predictive performance, and deploying the best-performing model into the integrated tracking system.

1) Walking speed prediction: The walking-speed dataset contains 1,274 participant records of participant characteristics including age, gender, body mass, body size, shoe size, and measured gait speed. The dataset was cleaned before model development by removing duplicated records, standardizing attribute names, encoding categorical gender information, and handling missing values. Missing values were processed using a two-stage imputation approach based on participant grouping.

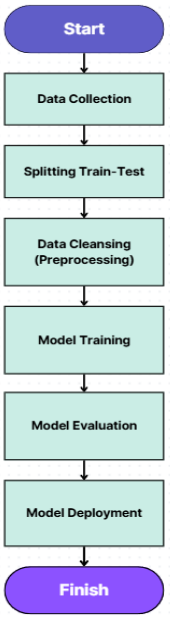


Fig. 3. Pipeline of machine learning process.

The processed dataset was divided into training and testing subsets using a random 80:20 train-test split, resulting in 1,019 records for training and 255 records for testing. Feature scaling was applied to the numerical inputs where required, with the scaler fitted using the training data to avoid information leakage into the testing set.

Three regression algorithms were evaluated using Ridge Regression, Random Forest Regression, and Gradient Boosting Regression. These models were selected to compare linear regression with ensemble tree-based approaches capable of representing nonlinear relationships between participant characteristics and walking speed.

2) Vehicle speed prediction: A separate vehicle travel dataset containing 146 trip records was used for vehicle-speed prediction. Unlike the human gait dataset, the vehicle dataset consists of only 146 trip records. A random 80:20 train-test split was initially performed, resulting in 116 records for training and 30 records for testing. Due to the relatively small number of training samples, a data augmentation technique was subsequently applied to expand the training set and provide additional samples for model development. The testing set was isolated before augmentation and remained completely untouched throughout the training process. The input features include travel distance, traffic condition, weather condition, vehicle type, day, and travel hour. These variables were selected to represent the environmental and temporal factors that may influence vehicle travel speed.

The dataset was preprocessed before training and divided into training and testing subsets. Three models were evaluated for vehicle-speed prediction using Random Forest Regression, Gradient Boosting Regression, and Long Short-Term Memory

(LSTM). Random Forest and Gradient Boosting were used as ensemble regression approaches, while LSTM was included to evaluate whether a recurrent neural network could capture patterns in the available travel data.

The models were evaluated using regression performance metrics, particularly MAE and the model achieved the lowest prediction error and was subsequently selected as the deployed vehicle-speed model. The selected model was serialized and integrated into the Flask backend together with the walking-speed model.

3) Evaluation metrics: In machine and deep learning, selecting appropriate evaluation metrics is important to measure the predictive accuracy of a trained model and to make fair comparisons across different algorithms. For regression tasks where the output is a continuous numerical value, three widely adopted metrics are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). Below are the equations used [18]:

a) Mean absolute error: MAE calculates the average errors between predicted and actual values without considering their direction in the dataset [18].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (3)$$

b) Root mean squared error: RMSE is the square root of MSE (Mean Squared Error); it measures on how far the variation in the dependent variable is explained by the independent variables. Lower RMSE indicates less error.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (4)$$

c) R-squared: R-squared is also known as the coefficient of determination which measures how well a model fits the data and the power to predict future outcomes. The ranges are capping from 0 to 1, where 1 indicates a perfect fit of the model to the data.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (5)$$

4) Prediction deployment: After evaluation, the selected models were exported using Joblib and stored in the Flask backend, avoiding the need for retraining during each journey. When a prediction is requested, the React dashboard sends the participant and route data to the Flask API. The backend selects the appropriate model based on transportation mode, processes the data and predicts travel speed. The predicted speed and route distance are then used to estimate travel time, calculate total journey duration, and determine delay risk against the checkpoint deadline.

This architecture separates data acquisition, storage, user interaction and prediction, allowing the RFID hardware to remain lightweight while machine learning processing is handled by the backend. It also allows models to be updated independently without modifying the physical RFID system.

IV. Result and Discussion

A. Result from RFID Reader

Fig. 4 shows the circuit setup of the proposed work. When an RFID tag is detected by the reader, the buzzer beside the ESP32 microcontroller will sounds to notify the user that the scan is detected, along with visual feedback on the LCD screen displaying "Scanned" to confirm the scanning action was successful; the status is updated to the current checkpoint.

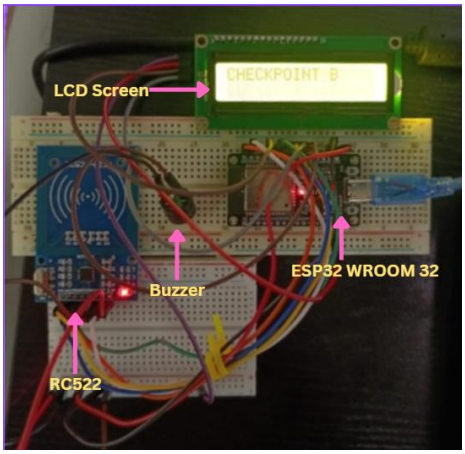


Fig. 4. Circuit setup.

Table I shows the scan detection test of the proposed work. From the results, the RFID reader can read the tag at a maximum of 4 cm from the reader. When the card tag is held further away from the reader, the card is not detected; no scan is triggered until the card enters the range of the reader scanning field. This is because the operating frequency of the RFID MFRC522 reader operates at high frequency of 13.56 MHz, which relies on inductive coupling for communication between the reader and the passive RFID tag. The magnetic field strength decreases rapidly as the distance increases, resulting in a relatively short operating range. Because passive RFID tags have no internal power source, they rely entirely on the reader's magnetic field to induce enough voltage in the tag antenna to power the chip.

TABLE I. Scan Detection Test

Test Case	Input	Expected Result	Pass/Fail
Card within range	RFID card held 1cm from reader	UID detected	Pass
Card at max range	RFID card held 2 cm from reader	UID detected	Pass
Card at max range	RFID card held 3 cm from reader	UID detected	Pass
Card at max range	RFID card held 4 cm from reader	UID detected	Pass
Card at max range	RFID card held 5 cm from reader	UID detected	Fail

As the magnetic field weakens rapidly with distance, the tag cannot retrieve sufficient energy to operate or send back its UID. This principle follows the equation of the inverse-cube law, where:

$$H \text{ is} \propto \frac{1}{d^3} \quad (4)$$

H= magnetic field strength

d= distance from the reader antenna

The greater the distance from the antenna, the weaker the magnetic field strength.

1) Tracking mode: Fig. 5 shows the serial monitor output when a member's RFID card was scanned at Checkpoint A. The ESP32 successfully initialized the RC522 RFID reader and established a Wi-Fi connection with the Supabase backend before entering tracking mode. When the RFID card was detected, the RC522 captured the card UID, and the ESP32 verified it against the registered member records in Supabase. The scanned UID was successfully matched to the registered member, CCY, and the system recorded the member as checked in at Checkpoint A. The ESP32 then updated the corresponding movement record in the Supabase database with the checkpoint and timestamp information.

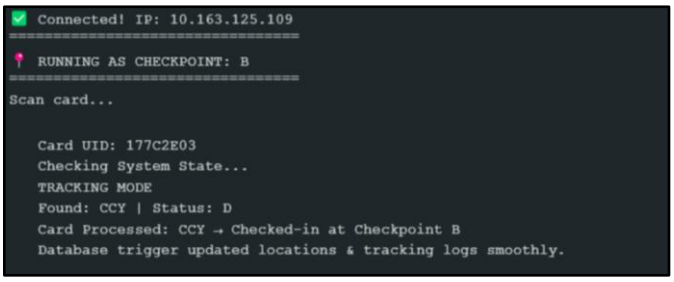


Fig. 5. Arduino serial monitor output during tracking scan.

The database update was propagated through Supabase Realtime WebSocket, allowing the React dashboard to update the member's status immediately. As shown in Fig. 6, the member's previous checkpoint status was changed from Checkpoint D (Queensbay Mall) to Checkpoint A (Universiti Malaysia Perlis), confirming successful RFID detection, database update, and real-time dashboard synchronization.



Fig. 6. Member status update to Checkpoint B in web dashboard.

2) Register mode: As shown in Fig. 7, an administrator can register or update a member's RFID card by entering the UID manually or selecting the Scan button. The system switches to REGISTER MODE, where the ESP32 captures the RFID UID and links it to the selected member's profile in the Supabase Profiles table. After successful registration, the system returns to TRACKING MODE, while the dashboard changes from "Waiting for Scan" to "Card Scanned", confirming successful registration.

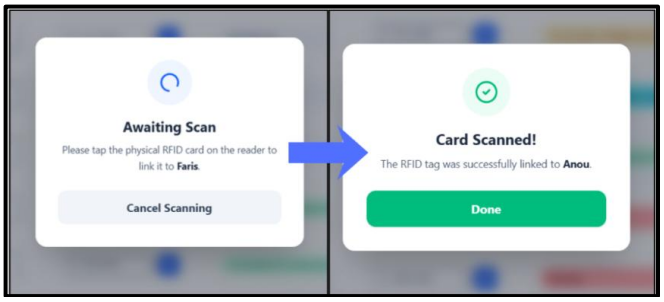


Fig. 7. After the card is scanned in the web dashboard.

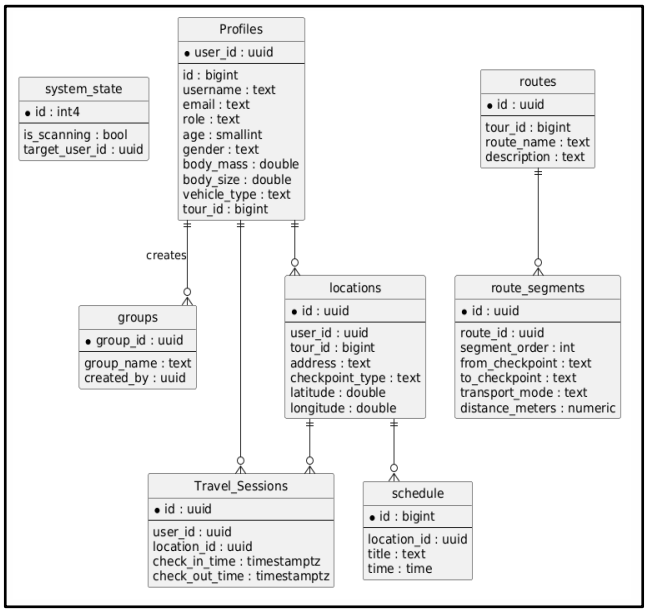


Fig. 8. Entity relationship diagram of RFID-based travel group tracking and monitoring system.

Fig. 8 shows the Entity Relationship Diagram (ERD) of the RFID-based Travel Group Tracking and Monitoring System. The Profiles table acts as the central entity, storing member information such as biometric data, RFID tag number and vehicle type. It is linked to the Groups, Travel_Session and Locations tables for member management, travel status, and location tracking. The Locations table records checkpoint scans, coordinates, timestamps and calculated distances to the exit checkpoint, supporting the Analytics dashboard and risk prediction. The Checkpoints table stores the location and exit status of each checkpoint. The Routes table defines each route, while Route_Segments stores its sequence, checkpoints, transportation mode, and travel distance. The Schedule table is linked to locations and stores planned activities and expected arrival times, which support system monitoring and risk prediction.

B. Supabase Table Relationship Design

C. Web Dashboard Test Result

Tables II, III, IV, and V show the member, group, and route analytics management page tests, respectively.

TABLE II. MEMBER MANAGEMENT PAGE TEST

Feature Tested	Expected Behaviour	Observed Behaviour	Result
RFID tag registration via scan button	New card UID links to selected member profile	Card UID written to database table and user interface shows output.	Pass
Manual RFID UID entry	Admin can type UID directly without scanning	UID saved to database on submit	Pass
Status edits by admin	Admin changes member status	Status updated in Supabase and UI refreshed	Pass
Status edits attempt by member role	Edit controls should be disabled for non-admin users	Edit fields not interactive for member role	Pass
Role toggle (Member to Admin)	Clicking action icon switches user role	Role updated and shield icon highlighted in red	Pass
Real-time update from external change	Table reflects changes made by another admin session	New status appeared without manual refresh	Pass

TABLE III. GROUP MANAGEMENT PAGE TEST

Feature Tested	Expected Behaviour	Observed Behaviour	Result
Create new group	New team card appears after naming	Team created and visible immediately	Pass
Assigning member to group	Selected member added to team list	Members appeared under correct team with inherited transport mode	Pass
Removed from group	Member unassigned and returned to pool	Member removed from team list instantly	Pass
Change group transport mode	All members in group update to new mode	Transport mode updated in bulk across all members	Pass
Rename group	Group name updates across all associated records	Name changed and reflected in member records	Pass

TABLE IV. ROUTE MANAGEMENT PAGE TEST

Feature Tested	Expected Behaviour	Observed Behaviour	Result
Create new route	Route saved with defined checkpoint sequence	Route appeared in route list with correct segments	Pass
Delete route	Route removed from list permanently	Route no longer appeared after deletion	Pass
Run risk prediction for member	Predicted travel time and risk percentage displayed	Segment breakdown and total journey time shown correctly	Pass
Segment transport mode variation	Each segment uses its assigned transport mode for prediction	Different predicted times shown per transport mode per segment	Pass

Across all four dashboard pages, Member Management, Group Management, Route Management, and Analytics, all test cases passed; Objective 2, which is functional webpages is achieved. Role-based access control, member and transport-mode updates, route-specific risk predictions, and transport-mode-based analytics were successfully verified. The Analytics page also correctly identified delayed members and triggered missing-member notifications, fulfilling the automated alert requirement.

TABLE V. ANALYTICS PAGES TEST

Feature Tested	Expected Behaviour	Observed Behaviour	Result
Display walking mode predictions	Members using walking mode show predicted time and risk	Risk percentages and time needed displayed per member	Pass
Display vehicle mode predictions	Members using vehicle mode show predicted speed and risk	Risk percentages and predicted speed displayed per member	Pass
Display bus mode predictions	Members using bus mode show predicted time and risk	Risk percentages displayed correctly for bus group	Pass
Toggle between transport types	Updates displayed member list and predictions	Member list and predictions changed per selected mode	Pass
Delay risk alert trigger	Member exceeding deadline threshold flagged as delay risk	Red highlight and extra time needed shown for at-risk member	Pass
Real-time prediction update	Predictions update when a new RFID scan occurs	Risk values refreshed after simulated new scan event	Pass

D. Result from Machine Learning

Fig. 9 shows risk prediction feature estimates whether a selected member can reach the exit checkpoint before the deadline. The user selects the route segments, transportation mode, and member before running the prediction. The system then uses trained machine learning models to estimate travel time for each segment based on the member's profile and selected transport mode. The results show the total travel time, remaining time, delay risk percentage, and status (On Track or Delay Risk). A segment breakdown also shows the predicted travel time and distance for each segment, helping administrators identify potential delays and adjust the route or transportation mode before the trip.

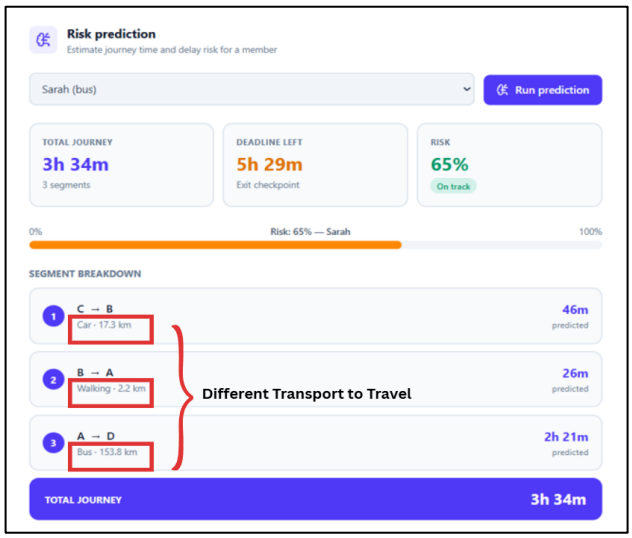


Fig. 9. Route time prediction.

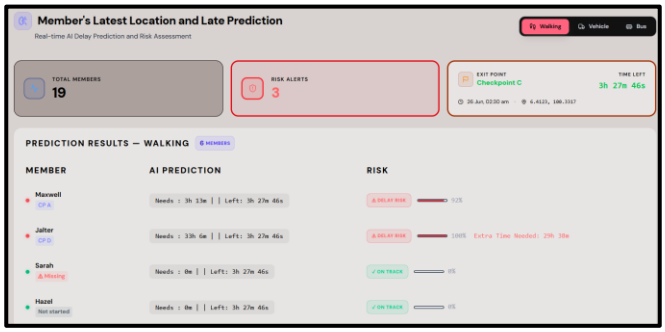


Fig. 10. Analytics page for transport using walking type member.

The Analytics Page integrates the trained machine learning models to provide real-time travel predictions. The system retrieves member biometric data, current location and remaining time from Supabase and sends them to the Flask backend for prediction. The dashboard displays each member's location status, predicted travel time, remaining time, and delay risk percentage using the formula:

$$\text{Risk} = (\text{Predicted Time Needed} \div \text{Time Remaining}) \times 100$$

Members predicted to exceed the deadline are marked as "DELAY RISK" and display the estimated extra time required. For example, in Fig. 10, three members are predicted to be late. The system updates predictions when new RFID scans or checkpoint deadline changes occur, allowing administrators to monitor members and act against potential delays.

TABLE VI. COMPARISON OF MODEL PERFORMANCE ON THE HUMAN DATASET

Metric	Random Forest	Gradient Boosting	Ridge Regression
R^2	0.9307	0.9313	0.0417
Mean Absolute Error (m/s)	0.0235 approx. (0.08 km/h)	0.0234 approx. (0.08 km/h)	0.0908 approx. (0.33 km/h)
Root Mean Square Error	0.0303	0.0302	0.1127
Training Time (s)	37.44	15.05	0.2003

As depicted in Table VI, Random Forest and Gradient Boosting achieved strong performance, with R^2 above 0.93, while Ridge Regression performed poorly with an R^2 of 0.0417 due to its inability to capture the non-linear relationship in gait speed.

Gradient Boosting achieved the lowest MAE of 0.0234 m/s and RMSE of 0.0302 m/s, closely matching Random Forest. It was also faster to train (15.05s vs. 37.44s), making it the preferred model. The low prediction error corresponds to approximately ±1.3 seconds of travel-time uncertainty per 100 m, indicating sufficient accuracy for reliable delay-risk assessment.

From the evaluation results, Gradient Boosting Regressor achieved the best result among these 3 models; hence, it is selected and downloaded to use as the model for the walking speed predictions and further implemented in the Flask backend later. The selected model and its scaler were then exported as a .pkl file and integrated into the Flask backend to make real-time walking speed prediction within the prediction system.

TABLE VII. COMPARISON OF MODEL PERFORMANCE ON THE VEHICLE DATASET

Metric	Random Forest	Gradient Boosting	LSTM
R^2	0.5177	0.6222	0.0296
Mean Absolute Error (km/h)	5.0695	5.0152	6.9337
Root Mean Square Error (km/s)	6.9913	6.1875	9.9186
Total Train Time (s)	26.25	23.42	8.95

As shown in Table VII, Gradient Boosting achieved the best performance, with the highest R^2 of 0.6222, lowest MAE of 5.0152 km/h, and lowest RMSE of 6.1875 km/h. Random Forest performed slightly worse, while LSTM had the lowest accuracy with an R^2 of 0.0296. Although LSTM had the shortest training time, Gradient Boosting was selected as the final model due to its better prediction accuracy and fit to the study situation.

The Gradient Boosting MAE represents approximately 32.09% relative error, which is higher than the walking-speed model due to factors such as traffic, road conditions, and driver behaviour.

Based on the evaluation of Random Forest, Gradient Boosting, and LSTM, the Gradient Boosting model was selected due to its lowest MAE and RMSE. The final model and scaler were saved as .pkl files and integrated into the Flask backend for speed estimation.

V. CONCLUSION

All three project objectives were successfully achieved. The RFID-based prototype successfully detected and registered tour group members at designated checkpoints, fulfilling the first objective. The second objective was achieved through the development of a real-time web dashboard that monitors member locations and provides automated missing-member alerts. The third objective was fulfilled through the training and evaluation of machine learning models for walking and vehicle speed prediction, with Gradient Boosting selected for its superior performance and integrated into the Flask backend. Overall, the integration of RFID, IoT devices, cloud databases and machine learning presented the feasibility of an automated tour group monitoring system. However, limitations such as the short reading range of HF RFID, Internet dependency, and limited dataset size remain challenges for large-scale deployment.

Future work will focus on upgrading the HF RFID module to a UHF long-range reader to achieve automatic scanning without requiring users to present their cards close to the reader. The system can also be improved by implementing stronger security and a multi-trip management structure. Finally, collecting larger and more diverse datasets across different locations, transportation modes and traffic conditions will improve the accuracy and generalizability of the machine learning models.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Universiti Malaysia Perlis (UniMAP) for providing the necessary facilities and funding to support this work.

REFERENCES

[1] "What is RFID technology – a definition and how it works," IoT Journey, Orange, 2025.

[2] R. Solutions, "The Basics of an RFID System," Atlas, 2025.

[3] "RFID readers collect data from RFID tags and labels," RFID4U, 2020.

[4] "NFC App to Manage Groups and Attendance," Global Tag, 2025.

[5] "Real-Time Monitoring," Digital Security Concepts, TourTrax, 2023.

[6] M. Farag and A. Abouelela, "Low-Cost Active Monitoring of Attendance using Passive RFID Technology," J. Ilmiah Teknik Elektro Komputer dan Informatika, vol. 8, no. 4, 2022, doi: 10.26555/jiteki.v8i4.25168.

[7] S. Sunder and A. Am, "Software development for asset management using RFID," 2025, doi: 10.37421/2168-9695.2025.14.309.

[8] M. Gopinath, S. Bodke, R. Shelake, and A. Shedge, "A Real Time Tracking System using Group Coordination," Zenodo, 2025, doi: 10.5281/zenodo.18114747.

[9] R. Belka, R. S. Deniziak, G. Łukawski, and P. Pięta, "BLE-Based Indoor Tracking System with Overlapping-Resistant IoT Solution for Tourism Applications," Sensors, vol. 21, no. 2, p. 329, 2021, doi: 10.3390/s21020329.

[10] D. P. Hutabarat and R. Saleh, "Human tracking system in a certain area based on IoT application," Int. J. Sci. Technol. Res., vol. 9, no. 4, 2020.

[11] A. Yadav, "RFID-Based Attendance Management System Using ESP32 and Google Sheets," ResearchGate, 2025, doi: 10.53032/tvcr/2025.v7n2.51.

[12] Pamith Madusanka Kumara Mehrdad Tahmasebi, and Devika Sethu, "Face Recognition based Attendance Management System," Int. J. Eng. Res. Technol., vol. 9, no. 5, 2020, doi: 10.17577/IJERTV9IS050861.

[13] S. S. Mahadik, R. Patil, and N. Patil, "Next-Gen Attendance Tracking: Leveraging Flutter and WiFi Connectivity for Educational Efficiency," ResearchGate, 2024.

[14] F. Ibezim, "Real-Time Web-Based Student Attendance System Utilizing Face Recognition and Supabase," ResearchGate, 2025, doi: 10.13140/RG.2.2.15090.00966.

[15] M. J. Abbass, R. Lis, and W. Rebizant, "A Predictive Model Using Long Short-Time Memory (LSTM) Technique for Power System Voltage Stability," Applied Sciences, vol. 14, no. 16, p. 7279, 2024, doi: 10.3390/app14167279.

[16] L. Wang, "RFID Integration with Internet of Things: Data Processing Algorithm Based on Convolutional Neural Network," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 6, pp. 812–820, 2025.

[17] R. Tomari, A. A. Kadir, W. N. W. Zakaria, D. Das, and M. B. Azni, "Deep Learning Model for Enhancing Automated Recycling Machine with Incentive Mechanisms," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 8, pp. 813–820, 2024.

[18] W.-M. Thor, "Regression Evaluation Metrics," APXML, 2023. [Online]. Available: https://apxml.com/courses/getting-started-with-scikit-learn/chapter-2-supervised-learning-regression/regression-evaluation-metrics.

Design and Formalization of a Distributed Intelligent Edge-Fog-Cloud Architecture for Industry 4.0 Systems

Hafsa AMALIK*, Hicham MEDROMI
Research Laboratory in Networks-Computer Science-Telecommunications & Multimedia (RITM)-System Architecture Team (EAS), National High School of Electricity and Mechanics (ENSEM), Hassan II University, Casablanca, Morocco

***Abstract*—Industry 4.0 systems increasingly require distributed decision-making capabilities to manage heterogeneous resources, large volumes of data, and time-sensitive operational constraints. This study proposes a distributed intelligent Edge-Fog-Cloud architecture combining a structural seven-component intelligent-entity model, a distributed cognition mechanism based on the Perceive, Analyze, Decide, and Act (PADA) cycle, hierarchical coordination, and graceful degradation. The contribution lies in the consistent integration of these mechanisms across Edge, Fog, and Cloud decision levels. The framework is illustrated through a proposed deployment scenario derived from the OCP Benguerir phosphate-processing facilities. The present study provides an architectural and qualitative analysis; it does not report physical deployment, prototype, simulation, or quantitative performance evaluation. Responsiveness, scalability, availability, and resilience are therefore treated as design objectives that require future empirical validation.**

***Keywords*—Industry 4.0; distributed intelligence; edge-fog-cloud architecture; cyber-physical systems; PADA; multi-agent systems; graceful degradation; IIoT**

I. INTRODUCTION

Industry 4.0 is reshaping industrial systems through the integration of Cyber-Physical Systems, Industrial Internet of Things technologies, and intelligent data-driven services [1], [2]. Modern industrial environments are now highly connected. They generate large volumes of operational data and demand decisions that are both faster and more adaptive. Traditional centralized architectures struggle to meet these requirements, particularly in terms of responsiveness, scalability, and resilience [3], [4].

Research on the digital transformation of industry has followed three main directions. The first builds on the convergence of Cyber-Physical Systems (CPS), the Industrial Internet of Things (IIoT), and cloud services [1], [2]. By tightly coupling physical assets with computational and communication capabilities, CPS provide a solid foundation for smart manufacturing. They enable real-time monitoring, process automation, and adaptive control [5], [6]. The later integration of cloud platforms and digital twin technologies extended these capabilities toward global supervision and long-term optimization [7], [8].

The second direction emerged as industrial environments grew more complex. Multi-Agent Systems (MAS) appeared as a natural answer to the limitations of centralized control. They offer distributed decision-making and cooperative behavior among heterogeneous resources [9], [10]. Despite this flexibility, their integration with IIoT infrastructures and their ability to sustain consistent multi-level coordination remain limited at industrial scale [10], [11].

These limitations motivated a third direction, based on Edge-Fog-Cloud architectures. Such architectures distribute computation across several layers. Processing is brought closer to physical assets, which reduces latency and improves scalability [3], [4], [12]. Their effectiveness has been demonstrated in predictive maintenance and smart manufacturing applications [8], [13]. However, most of these solutions remain centered on communication and data processing infrastructure. The organization of distributed intelligence, the formal representation of intelligent entities, and the integration of resilience mechanisms are rarely treated as first-class architectural concerns [5], [7]. This gap motivates the framework proposed in this work. A detailed positioning of the proposed architecture with respect to these approaches is provided in Section IV.

Recent studies have extended this continuum through collaborative task offloading, cloud-fog-edge reference frameworks, implementation-oriented industrial architectures, resilience analyses, and quantitative performance models [14], [15], [16], [17], [18]. These contributions confirm the importance of distributing computation and coordination across heterogeneous tiers. Nevertheless, explicit entity abstractions, cognition cycles, decision subsidiarity, and graceful degradation are not always integrated within a single architectural representation, which motivates the structural framework developed in this study.

This study proposes a distributed intelligent architecture built on four pillars: a formal model of intelligent entities, a distributed cognition mechanism based on the PADA cycle (Perceive, Analyze, Decide, Act), hierarchical coordination, and resilience through graceful degradation. The applicability of the framework is illustrated through a proposed deployment scenario derived from the OCP Benguerir phosphate-processing facilities. No physical deployment or quantitative validation is claimed.

The main contributions are: 1) a multi-layer Edge-Fog-Cloud architecture where intelligence is an intrinsic component; 2) a formal seven-component entity model; 3) a distributed PADA cognition model; 4) hierarchical coordination and

*Corresponding author.

resilience mechanisms; and 5) an industrially grounded deployment scenario based on the OCP Benguerir phosphate-processing facilities.

The remainder of the study is organized as follows. Section II presents the proposed architecture. Section III presents the industrial case study and the proposed deployment scenario. Section IV provides a comparative analysis with existing approaches. Section V discusses the findings, and Section VI concludes the study.

II. Proposed Distributed Intelligent Architecture

A. Design Principles

The proposed architecture is built around three design principles that address the limitations of existing Industry 4.0 frameworks [3], [4], [5].

The first principle is distributed responsibility, where perception, analysis, decision-making, and action capabilities are distributed among autonomous entities across all system levels. This organization improves responsiveness and reduces dependence on centralized infrastructures, which are increasingly unable to meet the real-time requirements of modern industrial environments [11].

The second principle is multi-level coordination, whereby local decisions remain aligned with broader operational objectives through structured coordination mechanisms. This approach preserves the flexibility required for local autonomy while ensuring consistency with system-wide goals [5].

The third principle is interoperability and modularity, through which the framework relies on standardized communication protocols and a modular deployment strategy. This facilitates progressive integration within heterogeneous industrial environments that typically combine legacy equipment with modern digital technologies [12].

Based on these principles, the architecture is organized into three complementary layers, namely Edge, Fog, and Cloud, which together provide the foundation for a distributed intelligent system capable of combining local responsiveness, coordinated decision-making, and operational resilience within a unified Industry 4.0 framework.

B. Multi-Level Edge-Fog-Cloud Architecture

The architecture proposed in this work adopts a multi-level Edge-Fog-Cloud organization designed to distribute intelligence according to the temporal and operational requirements of industrial decision-making. As illustrated in Fig. 1, the architecture is structured around three complementary layers that collectively support perception, analysis, coordination, and optimization functions across the industrial system.

This organization is motivated by the observation that industrial decisions differ significantly in terms of urgency, scope, and computational complexity. Certain events require immediate reactions at the equipment level, while others depend on a broader operational context or long-term analysis. A centralized architecture is often unable to efficiently address these heterogeneous requirements because all information must be transmitted to a common processing point before decisions can be made. In contrast, the proposed architecture distributes responsibilities across multiple decision horizons while maintaining global system coherence [3], [5].

The Edge layer represents the operational level of the architecture. It is directly connected to physical assets and industrial equipment through sensors and actuators. At this level, intelligent entities continuously monitor operational parameters such as vibration, current consumption, pressure, flow rate, conveyor speed, and product quality indicators. The primary objective of the Edge layer is to ensure real-time responsiveness by processing information locally and executing immediate corrective actions whenever necessary. This capability is particularly important in industrial environments where delays of only a few seconds may lead to equipment damage, production losses, or safety risks [3], [19].

The Fog layer acts as an intermediate coordination level between local operations and global supervision. Rather than focusing on individual equipment, Fog nodes maintain a contextual view of an entire production area. They aggregate information originating from multiple Edge entities, correlate events, execute predictive analytics functions, and coordinate operational decisions within their respective zones. In the proposed deployment scenario, three Fog nodes are mapped to the three major phases of the OCP Benguerir production line: destoning, screening, and loading. This organization allows operational decisions to benefit from a broader context while preserving the responsiveness required by industrial processes [4].

The Cloud layer provides strategic and system-wide intelligence. It is responsible for global supervision, long-term optimization, model management, and knowledge extraction from historical operational data. Unlike Edge and Fog layers, which primarily focus on immediate and medium-term decisions, the Cloud layer operates over longer time horizons and supports continuous improvement of the industrial system. It also coordinates distributed learning processes and maintains a global representation of production activities, equipment conditions, and operational performance [1].

A key characteristic of the proposed architecture is the existence of bidirectional information flows between the three layers. Operational data generated at the Edge level are progressively aggregated and contextualized as they move toward the Fog and Cloud layers. Conversely, optimization policies, model updates, and strategic decisions generated at higher levels are propagated downward and adapted to local operational conditions. This continuous exchange of information enables the architecture to combine local autonomy with global coordination.

Unlike many existing Edge-Fog-Cloud solutions that focus primarily on data processing infrastructure, the proposed architecture explicitly integrates decision-making, coordination, and resilience mechanisms within the architectural model itself. This approach transforms the architecture from a simple communication hierarchy into a distributed intelligent system capable of supporting complex Industry 4.0 applications while maintaining operational continuity and scalability.

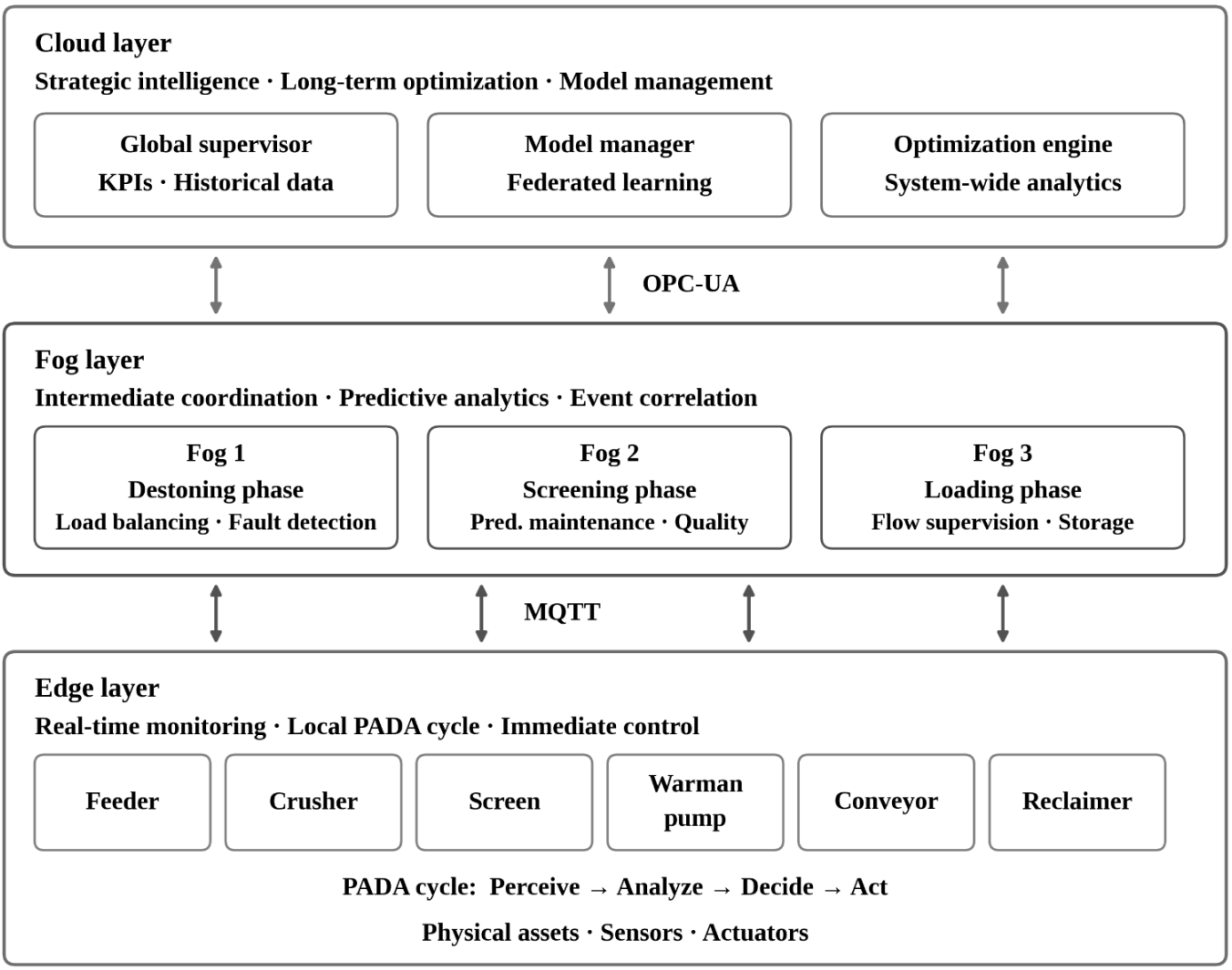


Fig. 1. Distributed edge-fog-cloud architecture.

C. Formal Modeling of Intelligent Entities

One of the limitations frequently observed in existing Industry 4.0 architectures concerns the lack of formal models capable of explicitly describing intelligent industrial entities and their role within distributed environments. Although concepts such as Cyber-Physical Systems, Industrial Internet of Things, and Multi-Agent Systems are widely adopted in the literature, the internal structure of intelligent components is often described at a conceptual level, making implementation, comparison, and reproducibility more difficult [5], [6]. This issue becomes particularly important in distributed architectures where multiple autonomous entities interact continuously across different decision layers.

To address this limitation, the proposed architecture introduces a formal model of intelligent entities that serves as the fundamental building block of the distributed intelligent system. The objective of this model is to provide a unified representation applicable to all components of the architecture, regardless of whether they operate at the Edge, Fog, or Cloud level. Such a representation facilitates system design, interoperability, scalability, and future extensions while maintaining a consistent architectural framework.

In the context of this work, an intelligent entity is defined as an autonomous component capable of perceiving its environment, processing information, making decisions, executing actions, and interacting with other entities through standardized communication mechanisms. This definition encompasses both physical and software components. For example, an Edge node attached to a feeder, a predictive maintenance service deployed at the Fog layer, and a federated learning module operating in the Cloud can all be represented using the same formal model.

Formally, each intelligent entity is represented by the following seven-component tuple:

$$E = (Id, P, S, D, A, C, I) \quad (1)$$

where, (Id) denotes the identity of the entity, (P) represents perception mechanisms, (S) defines the internal state, (D) corresponds to decision-making functions, (A) represents executable actions, (C) denotes communication capabilities, and

(I) describes integration constraints. As illustrated in Fig. 2, the proposed model organizes the behavior of each intelligent entity around seven complementary dimensions that collectively capture its operational, cognitive, and interactional characteristics. The model provides a common abstraction layer that can be applied consistently across the entire architecture.

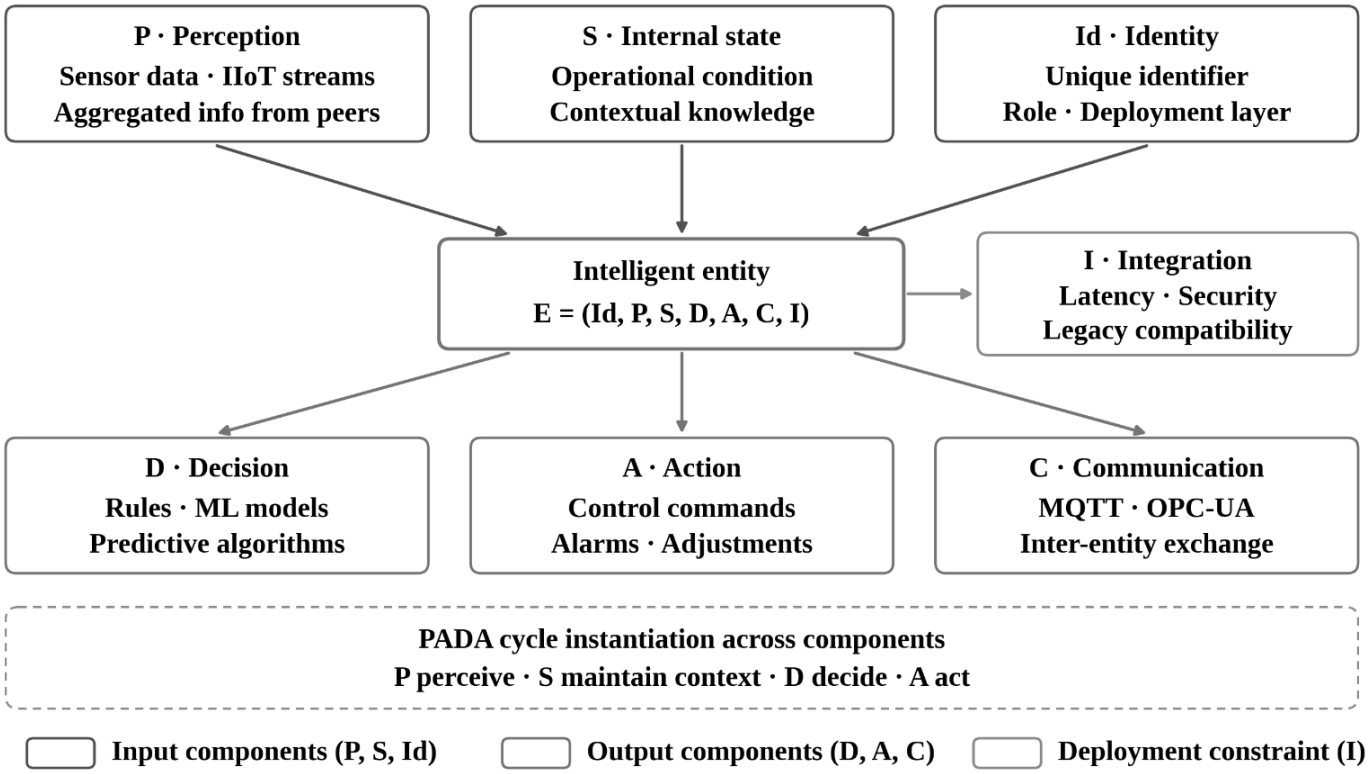


Fig. 2. Formal intelligent entity model.

The identity component ((Id)) uniquely identifies each entity within the architecture. Beyond simple identification, this component enables traceability, resource management, and coordination among distributed entities. It also facilitates the organization of entities according to their functional role and deployment layer.

The perception component ((P)) represents the mechanisms through which an entity acquires information from its environment. Depending on the architectural level, perception may involve direct acquisition of physical measurements from sensors, reception of processed data from neighboring entities, or access to aggregated information originating from higher layers. Perception constitutes the entry point of the intelligent decision cycle and provides the information required for subsequent analysis [19], [20].

The state component ((S)) captures the current operational condition of the entity. It includes both instantaneous observations and contextual information derived from previous interactions. Maintaining an internal state allows entities to preserve operational context and supports adaptive behavior in dynamic industrial environments.

The decision component ((D)) contains the mechanisms responsible for transforming information into decisions. These mechanisms may rely on predefined rules, machine learning models, predictive algorithms, or hybrid approaches depending on the complexity of the task and the computational resources available at each architectural layer. The explicit representation of decision capabilities is particularly important in distributed intelligent systems because it clarifies where and how decisions are generated [10].

The action component ((A)) defines the set of operations that an entity can perform once a decision has been produced. Actions may include local control commands, alarm generation, parameter adjustments, maintenance recommendations, or communication with other entities. Through this component, decisions are translated into observable effects on the industrial environment.

The communication component ((C)) represents the interfaces and protocols that enable information exchange among distributed entities. Communication plays a central role in maintaining coordination across Edge, Fog, and Cloud layers. It allows entities to share operational information, synchronize their activities, and contribute to collective decision-making processes [3], [4].

Finally, the integration component ((I)) captures the constraints that govern the deployment and operation of the entity within an industrial environment. These constraints may include latency requirements, computational limitations, security policies, interoperability standards, and compatibility with legacy industrial systems. By explicitly incorporating integration aspects into the model, the architecture addresses one of the major challenges of Industry 4.0 adoption, namely the

coexistence of modern intelligent services with existing industrial infrastructures [5], [19].

More precisely, Id is drawn from the set of unique entity identifiers; P maps observable inputs to perceived information; S belongs to the entity state space and contains current and retained context; D maps perceived information and state to an admissible decision; A maps a selected decision to an executable action; C defines the permitted messages, interfaces, and communication protocols; and I contains integration constraints, including latency, resource, interoperability, security, and safety requirements.

At the architectural level, entity behavior is represented as a transition system whose principal states correspond to Perceive, Analyze, Decide, Act, Degraded, and Recovery. A valid perception updates the entity state and enables analysis; an analysis result enables a decision when its confidence and validity conditions are satisfied; an authorized decision enables an admissible action; and completion or rejection of the action returns the entity to perception. Communication loss or node failure triggers a transition to Degraded, whereas restored connectivity triggers Recovery and controlled resynchronization before normal operation resumes.

The representation is governed by four architectural invariants: each entity has a unique identity; every executed action is traceable to an originating state and decision; an action must satisfy the constraints contained in I before execution; and loss of higher-level connectivity must not disable the locally designated critical functions. These invariants define requirements for a future executable specification. They are not presented here as mechanically verified properties.

The proposed model provides a consistent structural representation for intelligent entities participating in the architecture. By bringing perception, state management, decision-making, action execution, communication, and integration constraints into a single framework, it establishes a basis for distributed coordination across operational levels. The tuple does not constitute a complete formal verification framework: detailed data types, protocol semantics, executable transition guards, temporal properties, and proof obligations remain to be specified in implementation-oriented formalism.

D. Distributed Cognition Model

The effectiveness of a distributed intelligent architecture depends not only on the way computational resources are organized but also on how cognitive functions are distributed throughout the system. While many Industry 4.0 architectures incorporate advanced analytics, machine learning algorithms, or autonomous agents, the cognitive processes underlying decision-making often remain implicit. As a result, it becomes difficult to understand how information is transformed into actions and how decisions generated at different architectural levels remain coherent [5], [6], [21].

To address this challenge, the proposed architecture adopts a distributed cognition model based on the PADA cycle, which consists of four successive phases: Perceive, Analyze, Decide, and Act. The model provides a unified decision-making framework that can be instantiated across all intelligent entities regardless of their role or deployment layer. Rather than concentrating cognitive functions within a centralized controller, cognition is distributed among Edge, Fog, and Cloud entities according to their operational responsibilities and decision horizons.

The PADA model is directly connected to the formal intelligent entity model introduced in Section II-C. The perception component ((P)) supports information acquisition, the state component ((S)) maintains contextual knowledge, the decision component ((D)) generates decisions, and the action component ((A)) executes the selected responses. Through this integration, intelligent entities become autonomous cognitive units capable of interacting with their environment while remaining coordinated within the overall architecture.

The first phase of the cycle, Perceive, is responsible for acquiring information from the environment. Depending on the architectural level, perception may involve physical measurements collected from sensors, aggregated operational information received from neighboring entities, or global performance indicators obtained from higher-level services. The quality and relevance of the perceived information directly influence the effectiveness of subsequent cognitive processes.

The second phase, Analyze, transforms raw observations into meaningful knowledge. This phase may involve data filtering, event correlation, anomaly detection, predictive analytics, or machine learning inference. The objective is not merely to process information but to contextualize it and extract actionable insights. Several studies have emphasized the importance of contextual analysis in distributed industrial systems, particularly when multiple heterogeneous data sources must be combined to support decision-making [10], [12].

The third phase, Decide, generates an appropriate response according to the current state of the entity and the operational objectives of the system. Decisions may be derived from expert rules, optimization criteria, predictive models, or hybrid approaches. Because decision-making responsibilities are distributed across multiple architectural levels, decisions are generated locally whenever possible and escalated only when a broader operational context is required.

The final phase, Act, translates decisions into operational actions. Depending on the layer involved, actions may correspond to equipment control commands, alarm generation, maintenance recommendations, policy dissemination, or model updates. Through this phase, cognitive processes produce measurable effects on the industrial environment and contribute directly to system performance.

Fig. 3 illustrates how the PADA cycle is instantiated across the three architectural layers of the proposed framework.

PADA is conceptually related to established feedback and cognitive cycles. MAPE-K separates monitoring, analysis, planning, and execution around an explicit knowledge base [21]; OODA emphasizes rapid observation, orientation, decision, and action; and Sense-Plan-Act links perception to planning and control. PADA does not claim to replace these models or to introduce an entirely new feedback-loop principle. Its contribution in this work is architectural: the same Perceive-Analyze-Decide-Act structure is instantiated at Edge, Fog, and Cloud levels and coupled with decision subsidiarity, escalation,

and graceful degradation. Compared with MAPE-K, contextual knowledge is represented within the entity state S rather than as a separate shared component, while the Decide phase may include rule selection, planning, or optimization according to the layer concerned.

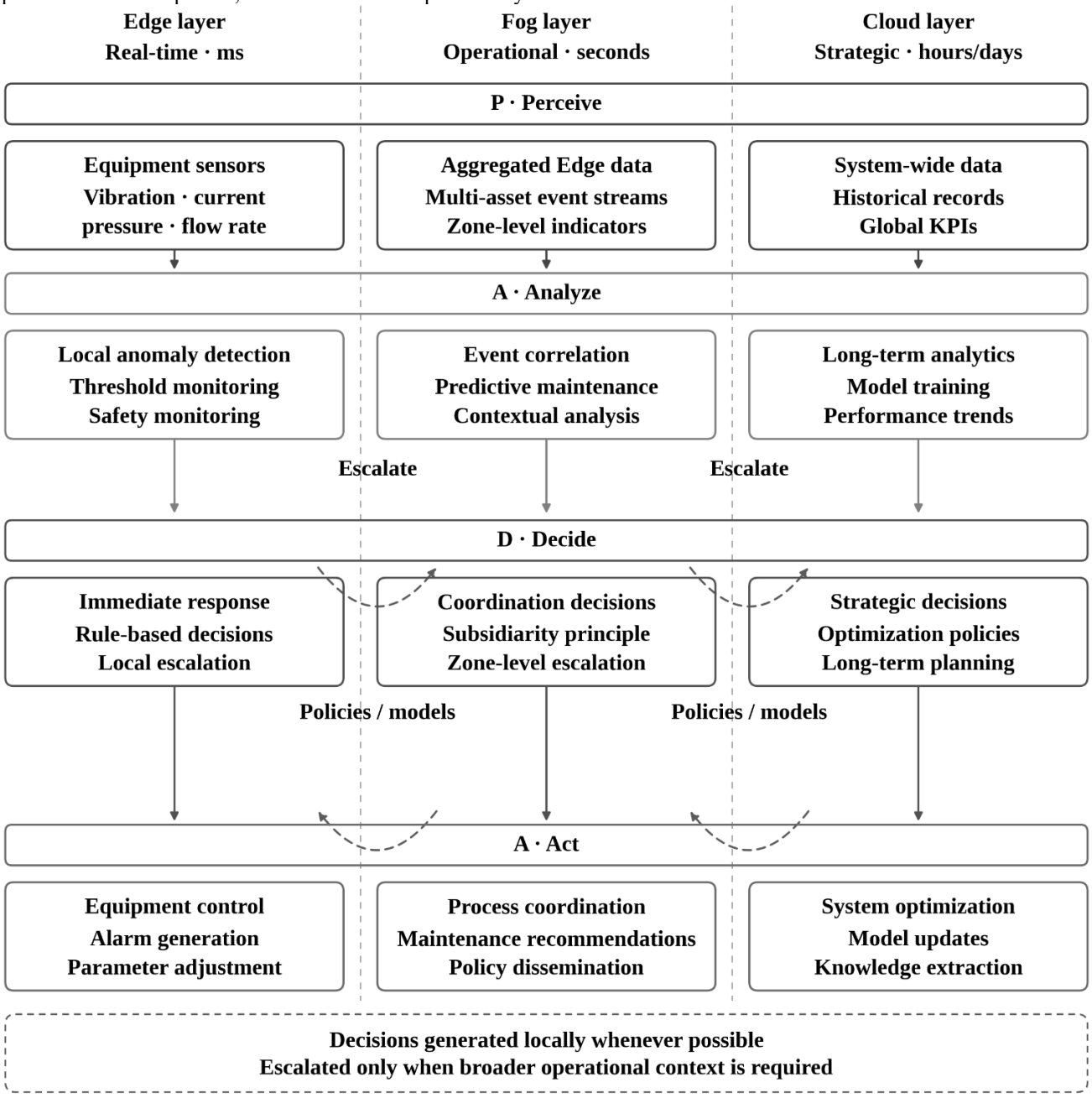


Fig. 3. PADA cycle distribution across the three-layer architecture.

The transition from one PADA phase to the next is conditional rather than purely sequential. Perception proceeds to analysis only when input validity requirements are met; analysis proceeds to decision when sufficient context or confidence is available; decision proceeds to action only when authorization, safety, and integration constraints are satisfied. Otherwise, the entity may retain the event, request additional context, escalate it to a higher layer, or enter a safe degraded state. These rules provide interaction semantics at the architectural level, while application-specific guards and thresholds remain implementation parameters.

A distinctive characteristic of the proposed approach is the distribution of cognition across multiple temporal horizons. At the Edge layer, the PADA cycle operates under strict real-time constraints and focuses on immediate operational events. Typical examples include anomaly detection, local safety monitoring, and equipment supervision.

At the Fog layer, cognition supports contextual analysis, predictive maintenance, event correlation, and operational coordination. Finally, at the Cloud layer, cognitive processes are oriented toward strategic optimization, knowledge extraction, model management, and long-term planning.

This hierarchical organization enables the architecture to combine local responsiveness with global consistency. Immediate decisions can be taken close to the physical process while strategic decisions benefit from a broader operational perspective. Consequently, the proposed architecture avoids the limitations of purely centralized systems while preserving coordination among distributed intelligent entities.

By introducing a common cognitive framework across all decision levels, the PADA model provides a coherent foundation for distributed intelligence, facilitating scalability, adaptability, and resilience in complex Industry 4.0 environments.

E. Coordination and Communication Mechanisms

The effectiveness of a distributed intelligent architecture depends not only on the capabilities of individual entities but also on their ability to exchange information and coordinate decisions. In Industry 4.0 environments, where cyber-physical systems, intelligent services, and connected devices operate simultaneously, maintaining consistency between local autonomy and global objectives becomes a major challenge [5], [6], [10].

To address this requirement, the proposed architecture adopts a hierarchical communication framework spanning the Edge, Fog, and Cloud layers. Rather than relying on centralized control, coordination is achieved through structured information exchanges and distributed decision-making processes.

Communication between Edge entities and Fog nodes is implemented through the MQTT protocol. Its lightweight publish–subscribe model makes it particularly suitable for Industrial Internet of Things environments characterized by resource-constrained devices and continuous data exchanges [10]. Edge entities publish operational data, alarms, and equipment states, while Fog nodes subscribe to the information relevant to their production area. This mechanism improves scalability and reduces communication overhead.

The Fog layer acts as the primary coordination level of the architecture. By aggregating information originating from multiple Edge entities, Fog nodes perform event correlation, predictive analytics, and operational supervision. This intermediate layer provides a broader operational context while preserving the responsiveness required for industrial processes [12], [13].

Communication between Fog and Cloud layers relies on OPC Unified Architecture (OPC-UA), a widely adopted standard for industrial interoperability and information exchange [22], [23]. Through OPC-UA, supervisory information, analytical models, and optimization policies can be exchanged in a structured and secure manner.

Coordination is further reinforced through decision subsidiarity. Local events are handled by Edge entities when the required information and authority are available, whereas decisions requiring a broader context or additional resources are escalated to Fog or Cloud levels [7], [24]. Each decision message is expected to carry its origin, scope, priority, timestamp, validity interval, and confidence. Safety-critical local actions take precedence over optimization requests; conflicting non-critical decisions are suspended and escalated to the responsible Fog node; and unresolved cross-zone conflicts are escalated to the Cloud supervisory level. These rules define a deterministic coordination policy rather than a distributed consensus protocol. To provide an implementation-oriented specification of this policy, Algorithm 1 describes local decision execution, priority-based conflict resolution, escalation, temporary Fog-level function takeover, and state reconciliation after connectivity restoration. The design of a distributed consensus protocol among equivalent Fog nodes remains outside the scope of this study.

Algorithm 1 specifies how each Edge entity processes observations and determines whether a decision can be executed locally or must be escalated. Local execution is permitted only when the decision satisfies the entity's authorization and safety constraints. When communication is unavailable, the entity activates its authorized degraded mode and buffers decisions and state updates. The procedure also defines deterministic priority rules for conflict resolution, temporary Fog-level takeover following an Edge-node failure, and state reconciliation after connectivity restoration. Although this specification improves the reproducibility of the coordination mechanism, its correctness and runtime performance remain to be evaluated through implementation and controlled experiments.

Algorithm 1: Distributed PADA Decision, Escalation, and Recovery Procedure

Initialize entity states S, decision rules R, authorization policies AP, priority levels PL, communication status CS, local state version V_E, and Fog state version V_F

Compute the initial connectivity and operational status of each Edge and Fog entity.

While the industrial system is operational, do

  For every Edge entity E_i do

    Acquire and validate observation o_i.

    Update internal state S_i using the validated observation.

    Execute the Perceive and Analyze phases of PADA.

    Compute candidate decision d_i using R and S_i

    Assign a priority according to safety, urgency, deadline, and operational impact.

    If d_i is locally authorized and satisfies the safety constraints, then

      Execute d_i at the Edge level.

      Record the action, timestamp, priority, and state version.

    Else if the Edge-Fog connection is available, then

Escalate d_i and its metadata to the associated Fog node.

If the Fog node detects conflicting decisions, then

Select the safety-critical decision.

Otherwise, select the decision with the highest priority.

When priorities are equal, select the earliest valid decision.

Escalate unresolved cross-zone conflicts to the Cloud

End if

Else

Activate the authorized degraded operating mode.

Buffer decisions and state updates with timestamps and sequence numbers.

End if

If E_i becomes unavailable, then

Reassign its critical monitoring functions to the associated Fog node.

Suspend functions that cannot be executed safely by the Fog node.

End if

If communication is restored, then

Transmit buffered records in sequence-number order.

Remove duplicate or invalid records.

Compare V_E with V_F

If a state conflict is detected, then

Preserve previously executed safety-critical actions.

Apply the most recent authorized consistent state.

Record the reconciliation result in the audit log.

End if

Resume normal PADA operation.

End if

End for

End while

A production implementation must complement this coordination policy with mutual authentication, role- or attribute-based authorization, encryption in transit, key management, audit logging, trust assessment, compromised-node detection, privacy controls, and safety interlocks. Fog-based security mechanisms can support distributed threat detection in large-scale IIoT environments [25]. These controls are architectural requirements associated with the integration component I; their implementation and security evaluation are outside the scope of the present study.

F. Resilience and Graceful Degradation

Resilience is a fundamental requirement for distributed industrial architectures, where communication failures, node outages, and infrastructure disruptions can compromise operational continuity [26]. Rather than assuming permanent availability of all system components, the proposed architecture adopts a graceful degradation strategy that preserves essential functions under partial failures. This approach is grounded in the distributed nature of the PADA cycle: since intelligent entities operate autonomously, they retain local perception, analysis, decision, and action capabilities even when connectivity with higher layers is interrupted [8], [26], [27].

Three degradation levels are considered. At the first level, Edge entities lose connectivity with their Fog node. In this case, they switch to autonomous mode, buffer operational data locally, and maintain critical functions such as anomaly

detection and safety monitoring until communication is restored. At the second level, Fog–Cloud connectivity is lost. Fog nodes continue operating using locally available information and previously received models, preserving event correlation, predictive maintenance, and coordination functions. Strategic optimization is temporarily suspended, but operational continuity is maintained [6], [8]. At the third level, a Fog node fails entirely. Neighboring Fog nodes dynamically assume coordination responsibilities for the affected zone, preventing isolated production areas [8], [26].

Once normal conditions are restored, the architecture initiates a progressive resynchronization process, redistributing buffered data, updated models, and optimization policies across all layers. Fig. 4 illustrates these three degradation levels and their corresponding recovery mechanisms.

The considered failure model covers communication-link loss and crash-stop unavailability of an Edge, Fog, or Cloud component; Byzantine behavior and compromised nodes are not modeled. Failures would be detected through configurable heartbeat timeouts, missed-message counts, and local health checks, with threshold values determined during implementation. Buffered records should carry timestamps, sequence numbers, and model or policy versions. Recovery should use idempotent replay, duplicate suppression, and version checks; when concurrent updates conflict, the responsible Fog node should retain safety-critical local actions and reconcile non-critical state according to the most recent valid policy. These mechanisms are specified as design requirements and have not yet been experimentally evaluated for consistency, recovery time, or availability.

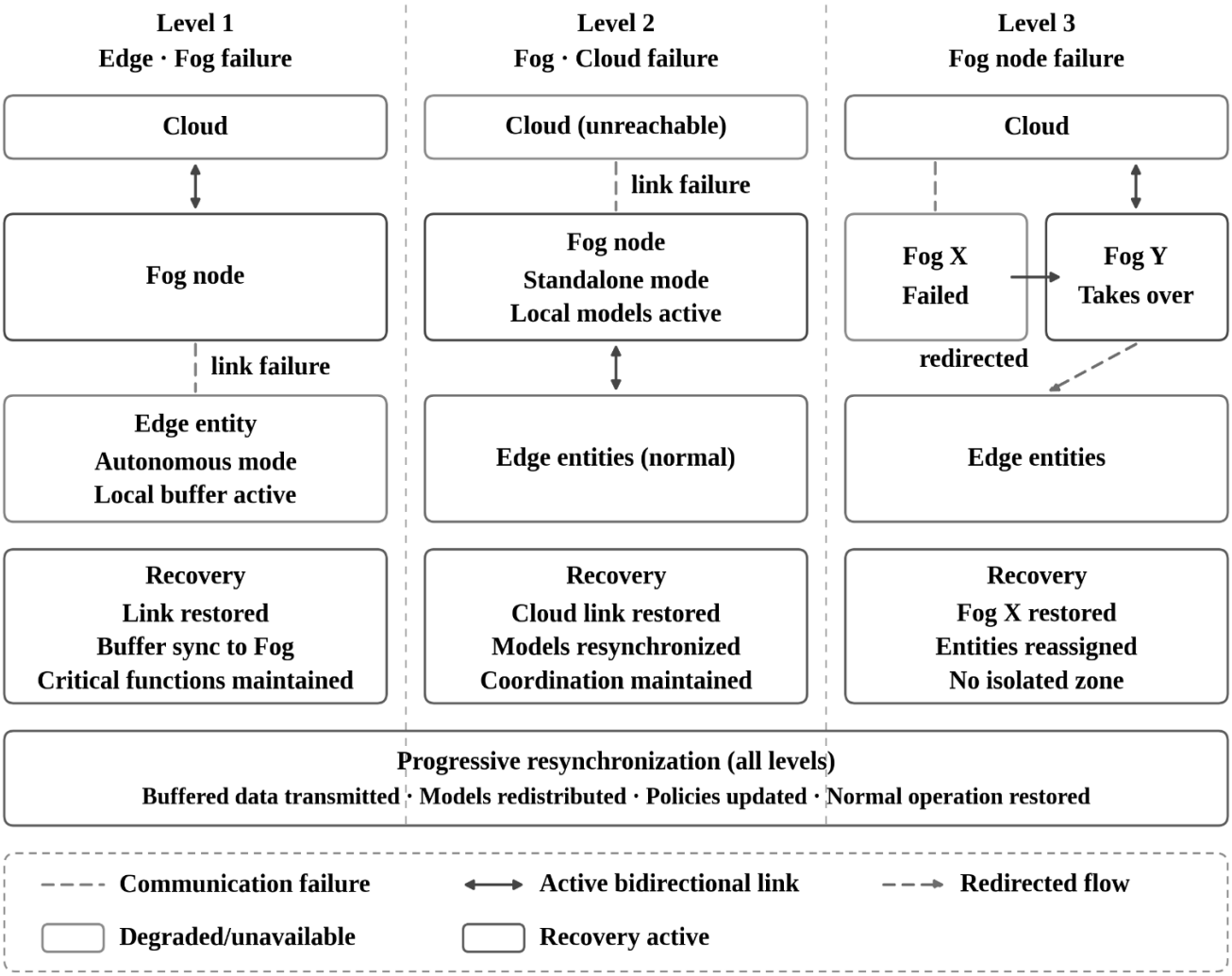


Fig. 4. Three-level graceful degradation and recovery mechanisms.

III. Industrial Case Study and Proposed Deployment Scenario

The fixed phosphate-processing facilities of the OCP Benguerir mining site are adopted as an industrial case study for defining the proposed deployment scenario. The architecture has not been physically deployed at the site, and this study does not use operational logs, field measurements, or observed system behavior as validation evidence. The production line is represented through three interconnected phases: destoning, screening, and loading. Their equipment, data sources, decision horizons, and continuity requirements provide an industrially grounded context for the architectural mapping [2], [5], [23], [28].

In the proposed mapping, Fog 1 would supervise feeders, crushers, and conveyors in the destoning phase, focusing on load balancing and fault detection. Fog 2 would coordinate screens, hydrocyclones, and Warman pumps in the screening phase, supporting predictive maintenance and quality monitoring [29]. Fog 3 would manage material-flow supervision and storage coordination during loading. This three-node organization follows the operational boundaries represented in the case study and is not a record of installed Fog infrastructure [8], [30].

In this scenario, intelligent Edge entities would be associated with the main assets of each phase and would execute local PADA cycles for monitoring, anomaly detection, and authorized immediate control. At the Cloud level, historical data and performance indicators would be consolidated for global supervision and long-term optimization. Table I summarizes this proposed mapping between production phases, equipment, and architectural components [5], [8], [31].

An illustrative data flow begins when a vibration sensor associated with a Warman pump produces a new observation. The Edge entity perceives and validates the measurement, updates its internal state, analyzes it against the configured anomaly rule or model, and either records a normal condition or issues an authorized local alarm. Events requiring wider context are published to Fog 2, which correlates them with other screening assets and may generate a maintenance recommendation. Relevant summaries can subsequently be transmitted to the Cloud for historical analysis and model management. This example explains the intended information flow; it is not an operational trace collected from the OCP site.

TABLE I. PROPOSED DEPLOYMENT OF INTELLIGENT ENTITIES FOR THE OCP BENGUERIR CASE STUDY

Production Phase	Main Equipment	Intelligent Edge Entities	Fog Node	Primary Functions
Destoning	Feeders, Crushers, Conveyors	Equipment-level entities	Fog 1	Monitoring, anomaly detection, local control
Screening	Screens, Hydrocyclones, Warman Pumps	Equipment-level entities	Fog 2	Predictive maintenance, coordination, quality monitoring
Loading	Conveyors, Reclaimers, Storage Silos	Equipment-level entities	Fog 3	Material flow supervision, operational coordination

IV. COMPARATIVE ANALYSIS

To position the proposed architecture within the current state of the art, this section compares it with three representative families of Industry 4.0 architectures identified in the literature, namely Cyber-Physical System (CPS)-based architectures, Multi-Agent System (MAS)-based architectures, and conventional Edge-Fog-Cloud architectures. The comparison is based on six descriptive dimensions: multi-level intelligence, entity representation, explicit cognition cycle, decision subsidiarity, graceful degradation, and illustration or validation basis. The comparison results are summarized in Table II.

CPS-based architectures [5], [23], [24] provide robust integration between physical assets and computational services, with a strong emphasis on connectivity, interoperability, and data management. However, they do not explicitly define a formal model for intelligent entities, nor do they distribute cognition across multiple architectural levels. MAS-based approaches [10], [11] naturally support distributed decision-making through autonomous agents but lack a unified representation applicable to heterogeneous industrial components deployed across Edge, Fog, and Cloud layers. Furthermore, large-scale coordination and integration with IIoT infrastructures remain challenging. Conventional Edge-Fog-Cloud architectures [3], [4], [13], [32] efficiently distribute computation to improve latency and scalability, but intelligence, decision-making, and resilience are generally implemented at the application level rather than being embedded into the architecture itself.

The proposed architecture combines a structural entity representation, a distributed PADA cycle, decision subsidiarity, and a conceptually specified degradation strategy. Table II indicates whether these capabilities are explicit, partial, or implementation-dependent within each architectural family. This qualitative comparison does not establish architectural completeness or performance superiority. Individual implementations may provide capabilities that are not captured by the selected dimensions.

TABLE II. COMPARISON OF REPRESENTATIVE INDUSTRY 4.0 ARCHITECTURES

Feature	CPS	MAS	Conventional EFC	Proposed framework
Multi-level intelligence	Partial	Supported	Partial	Explicit
Entity representation	Varies	Agent-based	Usually implicit	Seven-component model
Explicit cognition cycle	Usually implicit	Partial	No	PADA
Decision subsidiarity	Partial	Supported	Partial	Explicit
Graceful degradation	Implementation-specific	Implementation-specific	Partial	Conceptual
Illustration or validation basis	Varies	Varies	Varies	OCP conceptual case

V. DISCUSSION

The proposed architecture integrates intelligent entity representation, distributed cognition, hierarchical coordination, and graceful degradation within a unified Edge, Fog, and Cloud framework. The seven-component model clarifies the responsibilities of each entity, while PADA connects perception and internal state to decisions and actions. However, this architectural integration does not demonstrate better latency, availability, or scalability. These properties depend on the implementation, workload, network conditions, and available resources.

Distributing PADA across the three levels presents several trade-offs. Edge entities can process data close to industrial equipment and maintain selected functions during communication failures. This improves local autonomy but increases processing, storage, update, and safety requirements. At the Fog level, data aggregation provides a broader operational context. However, coordination and state synchronization may increase communication overhead as the number of entities grows. Recent studies confirm that task allocation and resource management influence the performance of collaborative Edge, Fog, and Cloud architectures [14], [15], [18].

The combined use of MQTT and OPC UA also introduces integration constraints. It may require protocol adaptation, shared information models, quality of service configuration, and duplicate message management. A physical implementation reported in [16] confirms the feasibility of combining local acquisition, preprocessing, and cloud transmission. Nevertheless, its experimental context differs from the industrial scenario considered in this work.

Decision subsidiarity creates a balance between local responsiveness and global consistency. An Edge entity may act before receiving a higher-level decision. Subsequent synchronization may then reveal conflicting states or policies. The proposed priorities, escalation rules, buffering, versioning, and resynchronization mechanisms provide an architectural basis for managing this issue.

Algorithm 1 specifies deterministic rules for conflict resolution, escalation, temporary Fog-level function takeover, and post-reconnection synchronization. Their implementation as executable software and the development of a distributed consensus protocol among equivalent Fog nodes remain future work.

Graceful degradation allows selected functions to continue during communication or node failures. This principle is consistent with existing work on fault-tolerant degradation and resilient distributed architectures [17], [27]. Its effectiveness must nevertheless be evaluated through failure injection and metrics such as availability, recovery time, message loss, and data consistency.

Security is also essential in distributed IIoT environments. Authentication, authorization, encryption, trust management, audit logging, and compromised node detection should be included in future implementation. Fog-based mechanisms can support distributed threat detection [25], but they also introduce processing and communication overhead.

The comparison in Table II is qualitative and does not demonstrate the superiority of the proposed architecture. Similarly, the OCP Benguerir case study represents a conceptual mapping rather than a physical deployment. No operational data, logs, or field measurements from OCP are used as validation evidence.

A. Limitations and Future Validation

This study is limited to architectural design, structural formalization, and qualitative analysis. Although the seven-component tuple defines the principal responsibilities of an intelligent entity, it does not yet constitute a complete executable or mechanically verified formal model. Detailed data types, temporal constraints, transition guards, proof obligations, and protocol semantics remain to be defined using an implementation-oriented formalism.

Algorithm 1 provides an implementation-oriented pseudocode specification of the PADA coordination and recovery mechanisms. However, it has not yet been implemented as executable software or evaluated under runtime conditions. The OCP Benguerir case represents a proposed architectural mapping without operational data, logs, or field measurements. Authentication, authorization, encryption, trust management, compromised-node detection, privacy protection, and industrial safety controls are identified as requirements but have not been implemented or evaluated. Consequently, the present results should not be interpreted as evidence of operational performance, availability, resilience, or scalability.

Validation should proceed in successive stages. The first stage should provide an executable specification of the entity transitions and architectural invariants. The second stage should implement a small-scale prototype or simulation of the distributed PADA cycle. The third stage should introduce controlled Edge-Fog, Fog-Cloud, and Fog-node failures. The evaluation should measure end-to-end latency, decision time, availability, recovery time, communication overhead, CPU and memory utilization, message loss, data consistency, and scalability under normal and injected-failure conditions.

A subsequent industrial pilot would then be required to compare experimental results with operational measurements and assess interoperability, safety, cybersecurity, and long-term reliability under field conditions.

VI. Conclusion

This study proposed a distributed intelligent Edge-Fog-Cloud architecture for Industry 4.0 systems. It integrates a structural seven-component entity representation, a distributed PADA cognition cycle, decision subsidiarity, and a three-level graceful degradation strategy. The contribution is architectural and qualitative. It organizes these mechanisms within a common representation across the Edge, Fog, and Cloud levels.

The seven-component tuple makes entity responsibilities and constraints explicit, while PADA links perceived information and internal states to decisions and actions at different temporal and operational horizons. The proposed transition rules, invariants, escalation metadata, and recovery requirements strengthen the specification. However, they do not constitute a complete executable or mechanically verified model.

The qualitative comparison positions the framework relative to broad CPS, MAS, and conventional Edge-Fog-Cloud architecture families without claiming universal superiority. Its main value lies in making entity representation, distributed cognition, decision subsidiarity, and graceful degradation explicit within a unified design. The associated resource requirements, coordination overhead, protocol integration, state reconciliation, and security costs remain to be quantified.

The OCP Benguerir case study illustrates a proposed mapping of intelligent entities to a realistic phosphate-processing context. It does not represent a physical deployment or provide operational evidence of performance. The applicability of the framework to other industrial domains must therefore be examined through implementation and comparative evaluation.

Future work will develop an executable prototype or simulation environment for the proposed architecture. The evaluation will implement conflict resolution, task reassignment, Fog-node takeover, post-reconnection synchronization, and security controls. It will measure latency, availability, recovery time, communication overhead, resource utilization, data consistency, and scalability under normal and injected-failure conditions. A subsequent industrial pilot will be required to assess interoperability, safety, and operational effectiveness under field conditions.

Declarations

The authors declare no competing financial interests or personal relationships that could have influenced this work.

References

[1] L. Da Xu, E. L. Xu, and L. Li, “Industry 4.0: State of the art and future trends,” Int. J. Prod. Res., vol. 56, no. 8, pp. 2941–2962, 2018, doi: 10.1080/00207543.2018.1444806.

[2] Y. Lu, “Industry 4.0: A survey on technologies, applications and open research issues,” J. Ind. Inf. Integr., vol. 6, pp. 1–10, Jun. 2017, doi: 10.1016/J.JII.2017.04.005.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop, Association for Computing Machinery, Aug. 2012, pp. 13–16. doi: 10.1145/2342509.2342513.

[5] L. Monostori et al., "Cyber-physical systems in manufacturing," CIRP Annals, vol. 65, no. 2, pp. 621–641, Jan. 2016, doi: 10.1016/J.CIRP.2016.06.005.

[6] D. G. S. Pivoto, L. F. F. de Almeida, R. da Rosa Righi, J. J. P. C. Rodrigues, A. B. Lugli, and A. M. Alberti, "Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review," J. Manuf. Syst., vol. 58, pp. 176–192, Jan. 2021, doi: 10.1016/J.JMSY.2020.11.017.

[7] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison," Engineering, vol. 5, no. 4, pp. 653–661, Aug. 2019, doi: 10.1016/j.eng.2019.01.014.

[8] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," Journal of Intelligent Manufacturing, vol. 31, pp. 127–182, 2020, doi: 10.1007/s10845-018-1433-8.

[9] T. Borangiu, D. Trentesaux, A. Thomas, P. Leitão, and J. Barata, "Digital transformation of manufacturing through cloud services and resource virtualization," Comput. Ind., vol. 108, pp. 150–162, Jun. 2019, doi: 10.1016/J.COMPIND.2019.01.006.

[10] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," Eng. Appl. Artif. Intell., vol. 22, no. 7, pp. 979–991, Oct. 2009, doi: 10.1016/j.engappai.2008.09.005.

[11] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart Agents in Industrial Cyber-Physical Systems," Proceedings of the IEEE, vol. 104, no. 5, pp. 1086–1101, May 2016, doi: 10.1109/JPROC.2016.2521931.

[12] A. Yousefpour et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, vol. 98, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.

[13] F. H. Khoso, A. Lakhan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, and K. Kanwar, "A microservice-based system for Industrial Internet of Things in fog-cloud assisted network," Engineering, Technology & Applied Science Research, vol. 11, no. 2, pp. 7029–7032, 2021, doi: 10.48084/etasr.4077.

[14] V. Barbosa et al., "Performance Evaluation of IoT-Based Industrial Automation Using Edge, Fog, and Cloud Architectures," Journal of Network and Systems Management, vol. 33, Art. no. 15, Jan. 2025, doi: 10.1007/s10922-024-09893-x.

[15] Z. Li et al., "A reference framework for the digital twin smart factory based on cloud-fog-edge computing collaboration," J. Intell. Manuf., vol. 36, no. 5, pp. 3625–3645, Jun. 2025, doi: 10.1007/s10845-024-02424-0.

[16] R. Dogea, X. T. Yan, and R. Millar, "Implementation of an edge-fog-cloud computing IoT architecture in aircraft components," MRS Commun., vol. 13, no. 3, pp. 416–424, Jun. 2023, doi: 10.1557/s43579-023-00364-z.

[17] M. Naiouf, E. Rucci, F. Chichizola, and L. De Giusti, Eds., Cloud Computing, Big Data & Emerging Topics, Communications in Computer and Information Science, vol. 1828. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-40942-4.

[18] Z. Lin et al., "Edge-fog-cloud hybrid collaborative computing solution with an improved parallel evolutionary strategy for enhancing tasks offloading efficiency in intelligent manufacturing workshops," J. Intell. Manuf., vol. 36, no. 7, pp. 4635–4662, Oct. 2025, doi: 10.1007/s10845-024-02463-7.

[19] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, 2014, doi: 10.1109/TII.2014.2300753.

[20] E. A. Lee, "Cyber-physical systems - Are computing foundations adequate?" position paper presented at the NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, TX, USA, Oct. 2006.

[21] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," Computer, vol. 36, no. 1, pp. 41–50, 2003, doi: 10.1109/MC.2003.1160055.

[22] W. Mahnke, S. H. Leitner, and M. Damm, OPC unified architecture. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-540-68899-0.

[23] A. W. Colombo et al., Eds., Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-05624-1.

[24] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," Manuf. Lett., vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.

[25] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi, "A survey of security in cloud, edge, and fog computing," Sensors, vol. 22, no. 3, Art. no. 927, 2022, doi: 10.3390/s22030927.

[26] S. Colabianchi, F. Costantino, G. Di Gravio, F. Nonino, and R. Patriarca, "Discussing resilience in the context of cyber-physical systems," Comput. Ind. Eng., vol. 160, p. 107534, Oct. 2021, doi: 10.1016/J.CIE.2021.107534.

[27] Y. Lin, S. Kulkarni, and A. Jhumka, "Automation of fault-tolerant graceful degradation," Distrib. Comput., vol. 32, no. 1, pp. 1–25, Feb. 2019, doi: 10.1007/s00446-017-0319-x.

[28] T. Zvarivadza et al., "On the impact of Industrial Internet of Things (IIoT) - mining sector perspectives," Int. J. Min. Reclam. Environ., vol. 38, no. 10, pp. 771–809, 2024, doi: 10.1080/17480930.2024.2347131.

[29] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," Mechanical Systems and Signal Processing, vol. 20, no. 7, pp. 1483–1510, 2006, doi: 10.1016/j.ymssp.2005.09.012.

[30] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "A critical review of smart manufacturing and Industry 4.0 maturity models: Implications for small and medium-sized enterprises," Journal of Manufacturing Systems, vol. 49, pp. 194–214, 2018, doi: 10.1016/j.jmsy.2018.10.005.

[31] V. Alcácer and V. Cruz-Machado, "Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems," Engineering Science and Technology, an International Journal, vol. 22, no. 3, pp. 899–919, Jun. 2019, doi: 10.1016/J.JESTCH.2019.01.006.

[32] S. Medina, D. Montezanti, L. Gómez D'Orazio, F. Garay, A. De Giusti, and M. Naiouf, "Distributed architectures based on edge computing, fog computing and end devices: A conceptual review incorporating resilience aspects," in Cloud Computing, Big Data & Emerging Topics, Communications in Computer and Information Science, vol. 1828, Cham, Switzerland: Springer, 2023, pp. 31–44, doi: 10.1007/978-3-031-40942-4_3.

A Vision-Based Error Compensation Method for Accurate Robotic Positioning in Automated Marking

Olzhas Olzhayev[1], Akhanseri Ikramov[2], Sayat Ibrayev[3]
Joldasbekov Institute of Mechanics and Engineering, Almaty, Kazakhstan[1, 2, 3]
International Information Technology University, Almaty, Kazakhstan[1]
Al-Farabi Kazakh National University, Almaty, Kazakhstan[2]

***Abstract*—Accurate robotic positioning is essential in automated marking because residual pose error directly shifts the permanent mark. This study presents a one-step vision-based residual error compensation method that measures the displacement between the target and the marking-tool position after the initial robot motion, transforms that displacement into the robot base frame, and applies it as a translational Tool Center Point (TCP) correction. The platform combines an AR4-MK3 six-degree-of-freedom manipulator, a fixed Intel RealSense D455 RGB-D camera, and a 50 W electromagnetic marking device. At five spatially distinct target points, the mean Euclidean positioning error decreased from 7.40 ± 1.35 mm to 2.14 ± 0.36 mm. The mean paired decrease was 5.26 mm (95% confidence interval: 4.04-6.48 mm), and all five points improved by 70.7%-71.4%. An exploratory point-level paired test gave $t(4) = 11.92$ and $p < 0.001$. The result supports the feasibility of inserting a local visual correction immediately before marking without changing the nominal robot kinematic model. The evaluation is a five-point pilot rather than a repeatability or full workspace study; raw repeated-trial, orientation-error, timing, robustness, and quantitative mark-quality measurements were not available. The conclusions are therefore limited to translational positioning under the tested camera and workcell configuration.**

***Keywords*—Vision-based error compensation; robotic positioning; automated marking; RGB-D camera; tool center point; industrial robotics; positioning accuracy**

I. Introduction

Industrial robots support welding, assembly, machining, inspection, material handling, and surface-processing operations because they combine programmable motion with high repeatability [1]. Repeatability, however, is not equivalent to absolute accuracy. Kinematic parameter errors, joint clearances, compliance, assembly tolerances, thermal effects, and environmental disturbances can displace the end-effector from a commanded Cartesian target [2]. These errors are important in marking because the final tool position determines the location of a permanent identifier.

Robot calibration and error compensation improve absolute positioning by identifying differences between nominal and actual robot behavior [3]. Kinematic calibration can reduce systematic geometric error, but non-kinematic effects such as compliance, thermal drift, payload, and base deformation may remain [4]. High-accuracy calibration also commonly uses laser trackers, coordinate-measuring machines, or photogrammetric systems, which improve traceability but increase workcell cost and integration effort [5].

Machine vision provides a complementary route because it observes the current relationship among the target, tool, and workpiece [6]. Vision-guided correction has been applied to machining, drilling, grinding, assembly, and navigation tasks [7]. In contrast to a purely global calibration model, a local measurement made after the initial robot motion can capture the net residual error produced by several upstream sources.

Prior work includes vision-guided robot localization [8], neural-network prediction of positioning error [9], and models that incorporate non-kinematic and weak rigid-base effects [10]. Hand-eye vision has also been used for indirect compensation [11], while photogrammetric calibration has improved camera-guided robot accuracy [12]. Recent studies additionally address uncertainty in TCP calibration [13], stereo-vision closed-loop measurement [14], and high-precision tool-path generation for engraving [15]. These approaches establish the value of external sensing, but many require global calibration, extensive data collection, iterative control, or specialized metrology.

Vision-based residual correction is well suited to automated marking. Even after the target is transformed into the robot coordinate system, the executed tool position can still contain errors from target localization, camera calibration, camera-to-robot registration, TCP definition, and manipulator mechanics. Because these errors accumulate along the measurement and positioning chain, the initially calculated TCP pose should not be treated as an error-free final command.

The proposed workflow inserts a single visual correction after the initial positioning motion and before tool activation. The camera measures the target-to-tool displacement; the displacement is expressed in the robot base frame, and the corresponding translational offset is added to the TCP command. The method does not replace robot or camera calibration; it compensates for the residual displacement that remains after those procedures.

The research gap is therefore application-specific rather than a claim that visual correction is new in itself. Existing studies mainly evaluate calibration, machining, drilling, assembly, or trajectory control. Automated marking requires target localization, frame transformation, residual error measurement, correction, and permanent mark execution to operate as one ordered sequence. A practical question is whether a simple pre-marking correction can improve a low-cost workcell without retraining a model or changing the nominal robot kinematics.

This study makes three contributions: 1) a marking-specific sense-correct-execute workflow that uses the same fixed RGB-D camera for initial localization and residual measurement; 2) an explicit frame-consistent formulation that transforms camera-space displacement before TCP correction; and 3) a transparent five-point pilot analysis that reports point-level dispersion, confidence intervals, and an exploratory paired comparison. The scope is translational correction under a fixed camera and fixed tool orientation. The study does not claim full six-degree-of-freedom compensation, industrial robustness, or metrological equivalence to a laser tracker.

II. Related Work

Industrial robot accuracy enhancement is commonly divided into kinematic calibration, extended error modeling, data-driven compensation, and external-sensor feedback. Kinematic methods identify geometric parameters from measured poses [16], whereas extended models incorporate compliance, backlash, base deformation, thermal effects, and load-dependent behavior [17]. Model-free methods learn commanded-to-measured error mappings [18], including neural, radial-basis, Gaussian-process, and other regression models [19]. Deep models can represent nonlinear workspace error [20], and similarity or incremental-learning strategies can reduce calibration effort [21]. These gains depend on the coverage and stability of the training data.

External metrology provides traceable end-effector measurements. Laser trackers are widely used for calibration and accuracy assessment [22], and real-time trajectory compensation can reduce path error [23]. Such systems offer higher accuracy than commodity RGB-D cameras but require line of sight, careful setup, and additional capital equipment. Visual guidance offers a lower-cost alternative for local correction. Examples include visually guided pose improvement in robotic machining [24], online visual correction for robotic drilling [25], and integrated machine-vision compensation architectures [26].

A reproducible visual correction chain requires both camera calibration and frame registration. Intrinsic calibration relates pixels and depth to camera-frame coordinates and is commonly estimated from multiple views of a calibration pattern [27]. Camera-to-robot registration is the hand-eye problem. Foundational solutions include the Tsai-Lenz method [28], separable rotation-translation estimation [29], Lie-group formulations of AX = XB [30], and dual-quaternion estimation [31]. Later work optimized pose selection [32], solved robot-world/hand-eye calibration jointly [33], and minimized reprojection error directly [34]. Reviews and comparative evaluations show that calibration accuracy depends on pose diversity, image localization error, and the selected solver [35]-[37].

The proposed correction is also related to image-based and position-based visual servoing. Classical visual servo control uses image or reconstructed pose error to generate motion commands [38]-[40], and modern robot-vision treatments emphasize explicit coordinate-frame conventions and stability assumptions [41], [42]. In multi-sensor systems, temporal and spatial calibration must be considered together when measurements and robot states are not synchronized [43]. Three-dimensional hand-eye calibration with line laser sensors [44] and learning-based vision for small and medium-sized production cells [45] further demonstrate the range of sensing and integration choices.

The Intel RealSense D455 is an active-stereo RGB-D device whose manufacturer specifies depth resolution up to 1280 x 720 and depth accuracy below 2% at 4 m under the stated test conditions [46]. Independent close-range characterization confirms that D455 performance varies with distance, surface geometry, and device-specific effects [47]. Evaluation under indoor mapping conditions also shows that measurement noise depends on range and surface inclination [48]. Consequently, a D455 cannot be treated as a substitute for a traceable metrology system, and its uncertainty forms part of the residual error reported in this study.

Published compensation results are not directly rankable because robots, workspaces, sensors, metrics, and test protocols differ. Accuracy and repeatability also change between offline and online robot-programming environments [49]. A laser-tracker and neural-network method reduced mean positioning error from 0.8497 to 0.0490 mm [50], while an offline deep-belief-network method reduced error from 0.469 to 0.084 mm [51]. Laser-tracker calibration has likewise demonstrated submillimeter improvement on industrial robots [52]. ISO 9283 provides standardized concepts for pose accuracy and repeatability [53], but the present pilot does not implement the complete standard protocol. Its contribution is the marking-specific local correction sequence, not superior absolute accuracy over these differently instrumented systems.

III. Materials and Methods

A. Experimental Vision-Guided Robotic Marking System

The experimental platform comprises an AR4-MK3 six-degree-of-freedom manipulator, a fixed Intel RealSense D455 RGB-D camera, a 50 W electromagnetic marking device, and a computer-based controller. The marking device is rigidly mounted on the robot end-effector. The camera observes the workpiece and tool from above in an eye-to-hand configuration. The camera was not moved during the reported five-point experiment.

The system uses two positioning stages. First, the target is localized from aligned color and depth data and transformed into the robot base frame to generate an initial TCP position. The robot then moves the marking tool to that command and comes to rest. Second, the vision system estimates the remaining target-to-tool displacement. After transforming the displacement into the base frame, the controller issues one corrected TCP command before activating the marking tool.

The controller performs image acquisition, target and tool localization, depth association, camera-to-base transformation, residual-error calculation, and robot command transmission. Fig. 1 identifies the interface view (left), the fixed camera and observed workpiece region (upper right), and the robot-mounted electromagnetic marking tool (lower right).

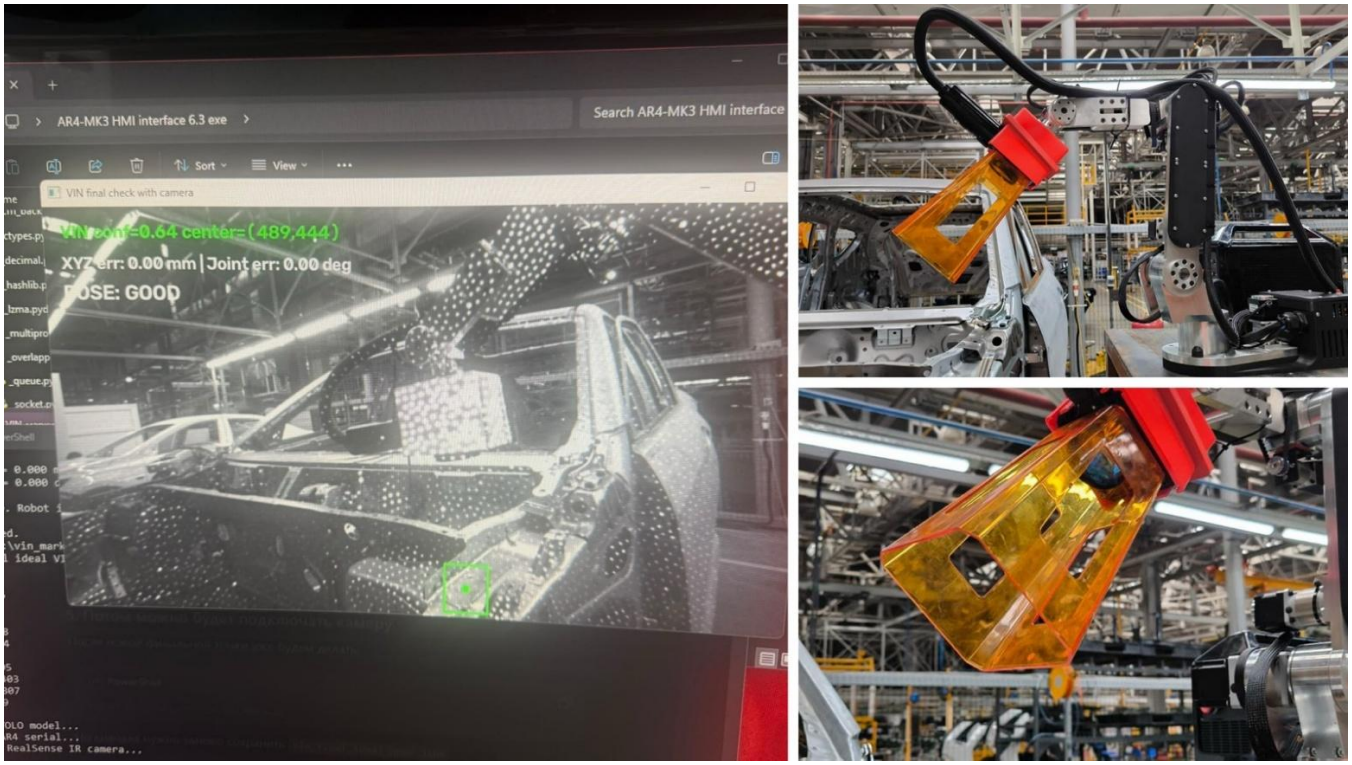


Fig. 1. Experimental vision-guided marking cell: camera and detection interface (left), fixed RGB-D camera and observed workpiece region (upper right), and AR4-MK3 manipulator with the electromagnetic marking tool (lower right).

B. Camera Calibration and Coordinate Frames

Four frames are used: the camera frame {C}, robot base frame {B}, TCP frame {T}, and workpiece/target frame {W}. The camera is fixed relative to the robot base. A camera-frame point p^c is mapped to the base frame by the homogeneous transformation $p^B = T\_BC\ p^c$, where T_BC contains the rotation R_BC and translation t_BC. The target and measured tool points are transformed independently before their difference is used for correction.

The intrinsic model deprojects an aligned pixel-depth observation according to $p^c = z\ K^{-1}\ [u, v, 1]^T$, where K is the camera intrinsic matrix and z is the depth associated with the detected image location. Target and tool image regions are localized by the workcell detector; the center of each detected region defines (u, v), and valid depth samples around that center are aggregated to suppress isolated depth outliers. This definition makes the visual measurement reproducible at the algorithmic level, although the detector version, exact acquisition resolution, calibration residuals, and per-frame raw observations were not retained in the experimental record.

The fixed eye-to-hand transformation T_BC was determined before the positioning experiment from paired camera-frame and robot-base observations and then held constant. The transformation must be validated in the actual work volume because intrinsic calibration error, target localization error, depth bias, and registration error propagate into the compensation vector. The original study did not archive an independent traceable uncertainty budget; consequently, the reported post-compensation error represents the combined robot-and-vision result rather than robot error alone.

C. Vision-Based Positioning Error Estimation

After the initial robot positioning, the camera observes the desired target and the marking-tool reference within the same scene. Let P_t and P_m denote their three-dimensional positions after deprojection and transformation into a common coordinate frame. The component-wise residual and Euclidean distance are defined in Eq. (1)-(7). In implementation, displacement E in {C} is rotated into {B} by R_BC; the translation t_BC cancels because E is a difference between two points.

Let the desired target position be defined as:

$$P_t = (x_t, y_t, z_t) \tag{1}$$

and the actual position of the marking tool after the initial robot movement as:

$$P_m = (x_m, y_m, z_m) \tag{2}$$

The positioning error along each coordinate axis is calculated as:

$$e_x = x_t - x_m \tag{3}$$

$$e_y = y_t - y_m \tag{4}$$

$$e_z = z_t - z_m \tag{5}$$

The resulting error vector is expressed as:

$$E = [e_x; e_y; e_z] \tag{6}$$

while the total positioning error is calculated using the Euclidean distance:

$$e = \mathrm{sqrt}\left(e_x^2 + e_y^2 + e_z^2\right) \quad (7)$$

The vector therefore represents both the magnitude and direction of the residual translation. Frame consistency is essential: a displacement measured in {C} cannot be added directly to a TCP position in {B}. Only the residual after rotation into {B} is passed to the compensation stage.

D. Proposed Error Compensation Method

The proposed method applies one translational correction after the initial motion. The initial TCP position is expressed in the robot base frame. The compensation offset is obtained from the camera-space residual through the rotation R_BC, and the corrected TCP position is transmitted as a second point-to-point command.

The initial TCP position is represented as:

$$P_{(TCP)} = \left[x_{(TCP)}; y_{(TCP)}; z_{(TCP)}\right] \quad (8)$$

Using the residual expressed in {C}, the base-frame compensation offset is defined as:

$$\Delta P^B = R_{BC} E^C \quad (9)$$

The corrected base-frame TCP position is then calculated as:

$$P_{TCP}^{corr} = P_{TCP} + \Delta P \quad (10)$$

or, in expanded base-frame form:

$$P_{TCP,corr}^B = [x_{TCP} + \Delta x^B, y_{TCP} + \Delta y^B, z_{TCP} + \Delta z^B]^T \quad (11)$$

The corrected position is transmitted to the robot controller only after the initial motion has settled. The tool orientation is held at the taught marking orientation and is not changed by the compensation algorithm. This design is appropriate for the planar workpiece used in the pilot, but it does not compensate for rotational error. Orientation error can change contact angle, normal force, and mark geometry and must be measured in a full six-degree-of-freedom extension.

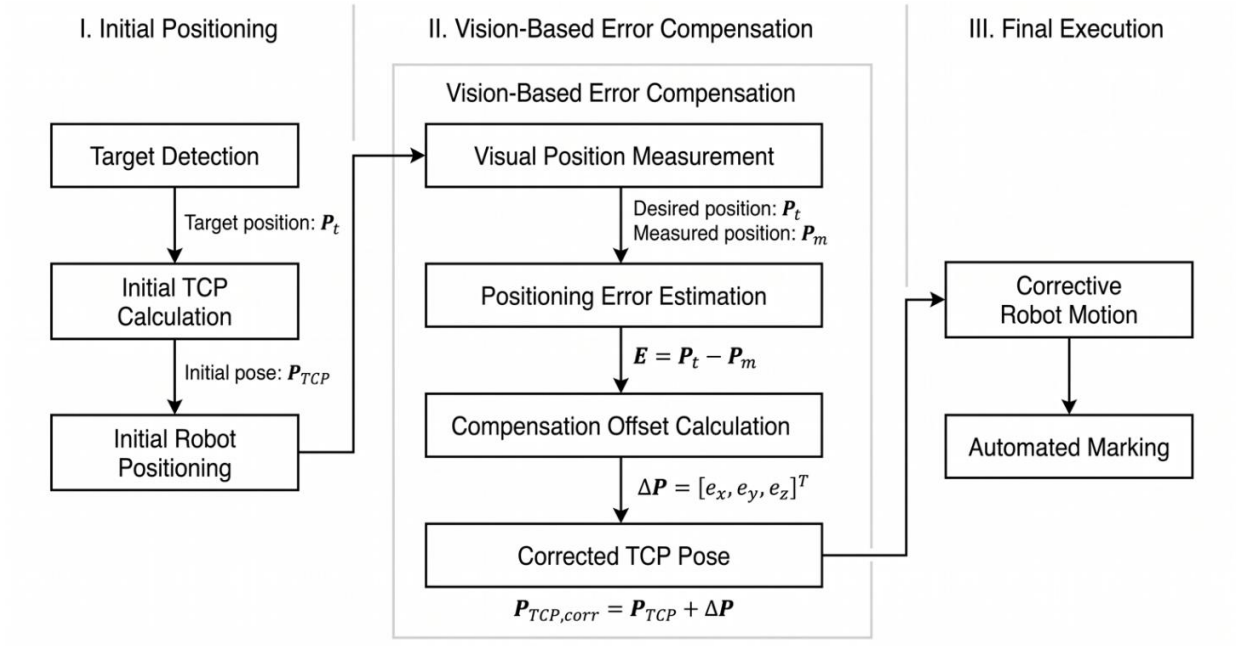


Fig. 2. Proposed vision-based error compensation framework for robotic positioning and automated marking.

Fig. 2 summarizes the marking-specific sequence. In the schematic, ΔP denotes the frame-consistent offset defined formally in Eq. (9). The first command uses the transformed target position, and the second uses the residual displacement measured after the robot has settled. The technical distinction from global calibration and learned workspace compensation is that the proposed offset is local, task-timed, and calculated immediately before permanent marking. The approach still depends on valid camera calibration and does not remove the need for metrological validation.

E. Experimental Procedure

The method was evaluated at five distinct target locations, P1-P5, within the accessible marking area. At each location, the robot executed the initial target command, stopped, and was observed by the D455. The Euclidean tool-to-target distance was recorded before compensation. The base-frame correction was then applied, the robot stopped again, and the final distance was recorded. The retained dataset contains one before-and-after pair for each spatial target point. Although preliminary workcell operation involved repeated positioning, trial-level raw measurements and exact base-frame coordinates for P1-P5 were not preserved; the manuscript therefore treats the five sites, not repeated trials, as the experimental units and does not claim ISO 9283 repeatability.

The primary outcome was Euclidean positioning error in millimeters. Secondary descriptive outcomes were the absolute paired decrease and percentage reduction at each point. The uncompensated first command serves as the within-point open-

loop baseline. No independent calibrated-positioning baseline, orientation-error series, cycle-time log, or robustness series for illumination, camera displacement, vibration, and workpiece rotation was available. These omissions are stated explicitly so that the pilot is not interpreted as a full industrial qualification study.

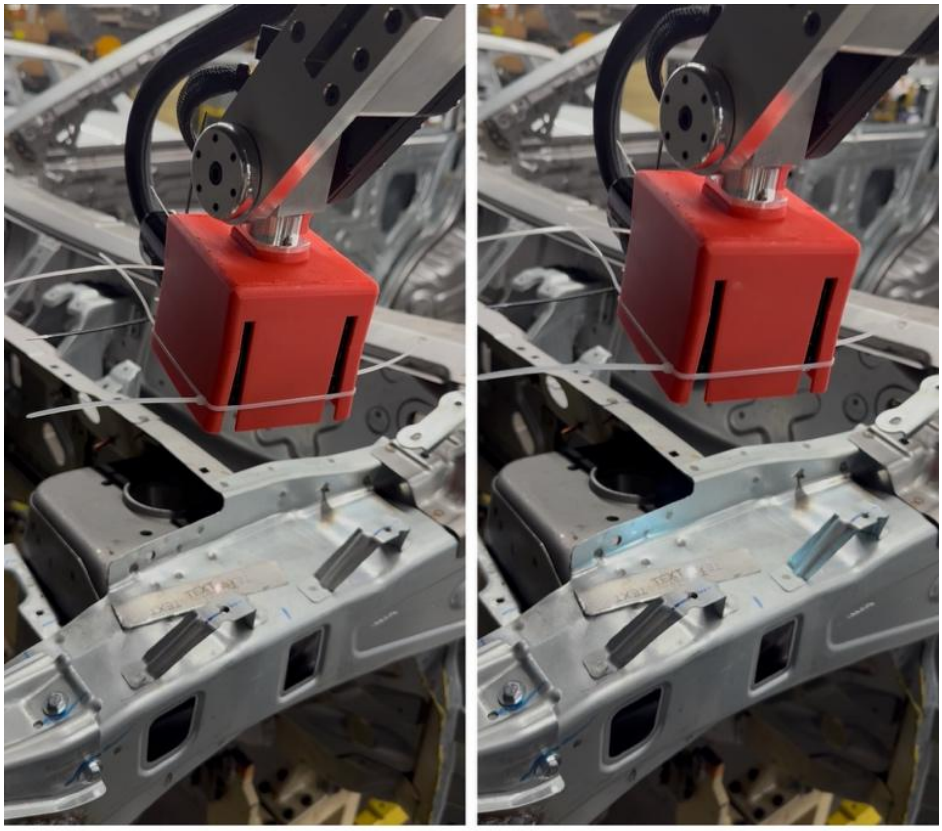
F. Statistical Analysis

Means, sample standard deviations, and two-sided 95% t confidence intervals were calculated across the five target points. A paired t-test compared the before- and after-compensation errors at the same spatial points, and Cohen's dz was calculated as the mean paired decrease divided by the standard deviation of the paired decreases. Because n = 5 and no repeated-trial data were retained, the inferential statistics are exploratory and characterize consistency across sampled locations rather than production repeatability.

IV. Results

Table I reports the five paired point-level observations. The uncompensated errors ranged from 5.8 to 9.1 mm, and the compensated errors ranged from 1.7 to 2.6 mm. Every target point improved, with percentage reductions between 70.7% and 71.4%. Fig. 3 illustrates the physical before-and-after correction sequence; the images are illustrative, whereas the numerical evaluation is based on the measurements in Tables I and II.

Fig. 3(a) shows the marking tool after the initial positioning command, and Fig. 3(b) shows the tool after the correction command. The corresponding dataset-level mean displacement decreased from 7.40 mm before compensation to 2.14 mm after compensation. Individual P1-P5 values are reported in Table I rather than overlaid on the photographs because the photographs do not provide a traceable scale for independent measurement.



a b

Fig. 3. Execution of the visual correction: (a) tool position after the initial command and (b) tool position after the correction command. Across P1-P5, the mean measured error decreased from 7.40 to 2.14 mm.

For target point i, percentage error reduction was calculated as:

$$R = \left(\frac{e_{(before)} - e_{(after)}}{e_{(before)}} \right) * 100\%$$

where, e_before,i and e_after,i are the Euclidean errors measured before and after compensation at the same target point. Table I preserves the individual observations.

The mean error was 7.40 ± 1.35 mm before compensation and 2.14 ± 0.36 mm after compensation. The mean paired decrease was 5.26 ± 0.99 mm, with a 95% confidence interval of 4.04-6.48 mm. The exploratory paired test yielded t(4) = 11.92, p = 0.0003, and Cohen's dz = 5.33. These statistics indicate a consistent reduction across the five sampled locations, but they do not substitute for repeated trials at each point.

TABLE I. POINT-LEVEL POSITIONING ERROR BEFORE AND AFTER VISUAL COMPENSATION

Target point	Before compensation (mm)	After compensation (mm)	Error reduction (%)
P1	7.2	2.1	70.8
P2	5.8	1.7	70.7
P3	8.4	2.4	71.4
P4	6.5	1.9	70.8
P5	9.1	2.6	71.4
Mean	7.4	2.14	71.0

TABLE II. DESCRIPTIVE AND EXPLORATORY POINT-LEVEL STATISTICS ACROSS FIVE TARGET LOCATIONS

Quantity	Estimate	95% CI
Before compensation	7.40 ± 1.35 mm	5.72-9.08 mm
After compensation	2.14 ± 0.36 mm	1.69-2.59 mm
Paired decrease	5.26 ± 0.99 mm	4.04-6.48 mm
Paired test	t(4) = 11.92	p = 0.0003
Effect size	Cohen's dz = 5.33	Point-level

After correction, the electromagnetic tool was activated. The photograph below shows one resulting mark and demonstrates process feasibility. It is not a quantitative mark-quality test: the retained experiment does not include independent measurements of mark-center deviation, character dimensions, depth, edge quality, or repeatability (Fig. 4). Accordingly, marking quality is not used as evidence for the positioning claim.

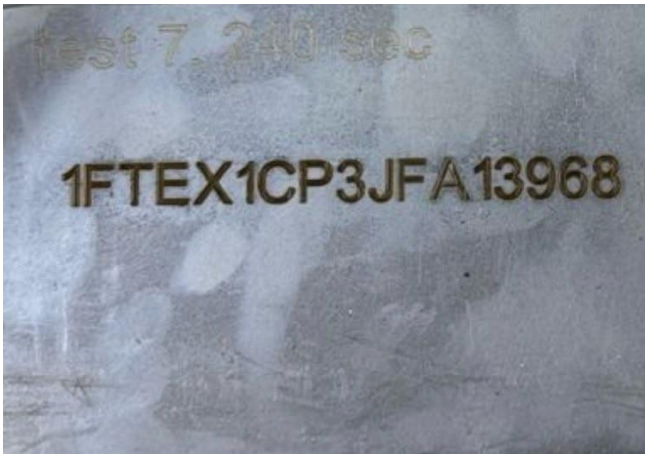


Fig. 4. Example mark produced after visual correction. The image demonstrates execution feasibility only; mark geometry was not measured quantitatively.

The available evidence therefore supports a narrow conclusion: one local translational correction reduced measured tool-to-target error at each of five tested sites. It does not establish orientation accuracy, camera-only measurement uncertainty, mark-quality repeatability, cycle-time overhead, or robustness to environmental changes.

V. DISCUSSION

Across the five target sites, compensation reduced mean error from 7.40 to 2.14 mm and produced a mean paired decrease of 5.26 mm. The narrow range of percentage reductions indicates similar relative behavior at the sampled sites. The point-level confidence interval excludes zero, and the exploratory paired test is consistent with a large reduction. Because only five site-level pairs were retained, these results should be interpreted as pilot evidence rather than an estimate of trial-to-trial repeatability.

The improvement is attributable to measuring the net residual displacement after the initial command. This timing allows one correction to incorporate the combined effect of localization, transformation, TCP, and manipulator errors present at that moment. The revised formulation also makes clear that camera-space displacement is rotated into the robot base frame before it is added to the TCP position. Direct addition without this transformation would be valid only if the camera and base axes were already coincident.

The approach reuses the camera already present in the marking cell, but this does not make the measurement metrologically independent. Published D455 characterizations show that depth error depends on range, surface inclination, and device behavior [47], [48]. The final 2.14 mm value therefore includes camera and registration uncertainty. The method should be described as low-cost visual compensation rather than as a replacement for traceable external metrology.

Table III positions the result against selected calibration and compensation studies. The numerical values are contextual because the studies use different robots, sensors, workspaces, and error definitions. Laser-tracker and learned models achieve smaller residuals but require calibration datasets and high-accuracy reference instruments. The proposed method uses one local RGB-D correction and addresses a less demanding marking setup. Its novelty is the integrated pre-marking sequence and explicit frame-consistent offset, not state-of-the-art absolute positioning accuracy.

TABLE III. CONTEXTUAL COMPARISON WITH SELECTED COMPENSATION METHODS

Study	Sensing and method	Reported result	Comparison notes
Ma et al. [10]	Laser tracker; non-kinematic and base-error model	Trajectory error -1.3231 to -0.1297 mm	Robotic milling; signed axis error
Jiang et al. [50]	Laser tracker; optimized neural network	Mean 0.8497 to 0.0490 mm	Global calibration and learning
Tao et al. [51]	Laser tracker; DE-DBN offline compensation	0.469 to 0.084 mm; 82.14%	Training data and traceable metrology
Li et al. [26]	Industrial camera, PLC, iterative correction	Accuracy within 0.231 mm	Different robot and repeated grasping task
Present pilot	D455 RGB-D; one local TCP correction	7.40 to 2.14 mm; 71.0%	Five marking sites; no trial-level repeats

The study has six principal limitations. First, the retained dataset contains only five spatial observations and no trial-level repetition series. Second, exact base-frame coordinates and robot joint configurations for P1-P5 were not archived, which limits workspace reproducibility. Third, the work uses one camera, robot, tool, and controlled workcell. Fourth, illumination, camera displacement, workpiece rotation, vibration, and larger workspace changes were not tested independently. Fifth, only translation was corrected; orientation and contact-normal error were not measured. Sixth, image-processing time, error-estimation time, correction-motion time, total added cycle time, and quantitative mark quality were not instrumented. These constraints preclude broad industrial generalization. A follow-up study should use an ISO 9283-informed repeated-point protocol, publish raw trials and target coordinates, validate camera and hand-eye uncertainty, add calibrated and alternative vision baselines, measure timing and mark geometry, and test environmental perturbations.

VI. Conclusion

This study presented a marking-specific, vision-based residual error correction for an AR4-MK3 robot. The method measures the target-to-tool displacement after the initial command, rotates that displacement from the camera frame into the robot base frame, and applies one translational TCP correction before marking. Across five target sites, mean error decreased from 7.40 ± 1.35 mm to 2.14 ± 0.36 mm. The mean paired decrease was 5.26 mm (95% confidence interval: 4.04-6.48 mm), and all sites improved by approximately 71%.

The result demonstrates feasibility under the tested configuration, not complete industrial validation. The study does not establish repeated-trial precision, rotational accuracy, camera metrological uncertainty, mark-quality repeatability, processing time, or robustness to environmental changes. Future work should report raw repeated measurements at documented workspace coordinates, compare calibrated and alternative vision baselines, quantify the full camera-to-robot uncertainty chain, measure orientation and mark geometry, and evaluate cycle time and robustness before broader deployment claims are made.

Acknowledgment

This study was funded by the research project “BR24992947 - Development of Robots, Scientific, Technical, and Software for Flexible Robotization and Industrial Automation (RPA) in Automotive Industrial Enterprises in Kazakhstan Using Artificial Intelligence”.

References

[1] Phuyal, S., Bista, D., & Bista, R. (2020). Challenges, opportunities and future directions of smart manufacturing: a state of the art review. *Sustainable Futures*, *2*, 100023.

[2] Wu, J., & Tao, Y. (2020). Review on the research status of positioning accuracy reliability of industrial robots. China Mechanical Engineering, 31(18), 2180-2188.

[3] Li, Z., Li, S., & Luo, X. (2021). An overview of calibration technology of industrial robots. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 23-36.

[4] Liao, W. H., Tian, W., Li, B., Li, P. C., Zhang, W., & Li, Y. F. (2022). Error compensation technology and its application progress of an industrial robot. *Acta Aeronautica et Astronautica Sinica*, *43*(5), 627142.

[5] Santolaria, J., Conte, J., & Ginés, M. (2013). Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector. *The International Journal of Advanced Manufacturing Technology*, *66*(9), 2087-2106.

[6] Bai, Y., Zhang, B., Xu, N., Zhou, J., Shi, J., & Diao, Z. (2023). Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review. *Computers and Electronics in Agriculture*, *205*, 107584.

[7] Ding, T., Ouyang, P., Zhang, H., Feng, X., Hua, L., & Zhu, D. (2025). 3D vision-guided robotic grinding framework for repairing random defects. *IEEE Transactions on Automation Science and Engineering*, *22*, 15819-15831.

[8] Olzhayev, O., Tursynova, A., & Ibrayev, S. (2026). A Coordinate Transformation Framework for Vision-Guided Robotic Positioning in Automated Vehicle Body Marking. International Journal of Advanced Computer Science & Applications, 17(8), 568.

[9] Hu, J., Hua, F., & Tian, W. (2020, March). Robot positioning error compensation method based on deep neural network. In *Journal of Physics: Conference Series* (Vol. 1487, No. 1, p. 012045). IOP Publishing.

[10] Ma, S., Deng, K., Lu, Y., & Xu, X. (2023). Error compensation method of industrial robots considering non-kinematic and weak rigid base errors. *Precision Engineering*, *82*, 304-315.

[11] Olzhayev, O., Imanbayeva, N., Mamikov, S., & Baibek, B. (2025). A Novel YOLO-Like Multi-Branch Architecture for Accurate Apple Detection and Segmentation Under Orchard Constraints. International Journal of Advanced Computer Science and Applications, 16(11).

[12] Ulrich, M., Steger, C., Butsch, F., & Liebe, M. (2024). Vision-guided robot calibration using photogrammetric methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, *218*, 645-662.

[13] Li, Y., Li, B., Zhao, X., Cheng, S., Zhang, W., & Tian, W. (2023). Error similarity analysis and error compensation of industrial robots with uncertainties of TCP calibration. *Applied Sciences*, *13*(4), 2722.

[14] Zhang, S., & Qiao, Z. (2025). An autonomous calibration method for end-effector positioning error in robots using stereo vision closed-loop measurement. *The International Journal of Advanced Manufacturing Technology*, *139*(7), 3511-3523.

[15] Adamík, M., Babinec, A., Dekan, M., & Kovaríková, Z. (2025, October). 6 DoF Robot Tool Path Generator for the Purpose of High-Precision Engraving of Unknown Surfaces. In *International Symposium on Measurements and Control in Robotics* (pp. 3-16). Cham: Springer Nature Switzerland.

[16] Chen, T., Yang, W., Li, S., & Luo, X. (2025). Data-driven calibration of industrial robots: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, *12*(8), 1544-1567.

[17] Saleh, R. A., Nasser, S. F., & Abdullah, T. S. (2025, August). Advances in error compensation for robotic manipulators: a systematic review from geometric calibration to AI-driven hybrid control. In *2025 5th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*(pp. 1-10). IEEE.

[18] Wang, Z., Chen, J., San, H., Gong, M., Zhang, X., & Chen, W. (2025). Kinematic calibration of delta robots via model-free PSO-RBF prediction and compensation. *Measurement*, 118828.

[19] Romano, G., Bilancia, P., Locatelli, A., Mucciarini, M., Iori, M., & Pellicciari, M. (2026). A machine learning–based tool for enhancing position accuracy in industrial robots with a reduced dataset. *Robotics and Computer-Integrated Manufacturing*, *101*, 103289.

[20] Che, C., Liu, B., Li, S., Huang, J., & Hu, H. (2023). Deep learning for precise robot position prediction in logistics. *Journal of Theory and Practice of Engineering Science*, *3*(10), 36-41.

[21] Raghunath, M. P., Deshmukh, S., Chaudhari, P., Bangare, S. L., Kasat, K., Awasthy, M., ... & Waghulde, R. R. (2025). PCA- and PSO-based optimized support vector machine for efficient intrusion detection in the Internet of Things. Measurement: Sensors, 37, 101806.

[22] Wang, S., Yang, J., & Sang, X. (2026). Laser-Tracker-Based Robot Pose Measurement Using PSD Spot Sensing and Multi-Sensor Fusion with Simulation Validation. *Micromachines*, *17*(3), 290.

[23] Omarov, B., Omarov, B., Rakhymzhanov, A., Niyazov, A., Sultan, D., & Baikuvekov, M. (2024). Development of an artificial intelligence-

enabled non-invasive digital stethoscope for monitoring the heart condition of athletes in real-time. Retos, 60, 1169-1180.

[24] Momynkulov, Z., Tursynova, A., Olzhayev, O., Ikramov, A., Ibrayev, S., & Omarov, B. (2025). Three-dimensional trajectory planning for robotic manipulators using model predictive control and point cloud optimization. Computer Modeling in Engineering & Sciences, 145(1), 891-918.

[25] Maghami, A., & Khoshdarregi, M. (2024). Vision-based target localization and online error correction for high-precision robotic drilling. *Robotica*, *42*(9), 3019-3043.

[26] Li, Y., Lai, P., Yang, T., & Li, Z. (2026). Design and development of a machine-vision-based positioning compensation system for industrial robots. Discover Applied Sciences, 8, 174. https://doi.org/10.1007/s42452-025-08196-3

[27] Zhang, Z. (2000). A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1330-1334. https://doi.org/10.1109/34.888718

[28] Tsai, R. Y., & Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Transactions on Robotics and Automation, 5(3), 345-358. https://doi.org/10.1109/70.34770

[29] Horaud, R., & Dornaika, F. (1995). Hand-eye calibration. The International Journal of Robotics Research, 14(3), 195-210. https://doi.org/10.1177/027836499501400301

[30] Park, F. C., & Martin, B. J. (1994). Robot sensor calibration: Solving AX = XB on the Euclidean group. IEEE Transactions on Robotics and Automation, 10(5), 717-721. https://doi.org/10.1109/70.326576

[31] Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. The International Journal of Robotics Research, 18(3), 286-298. https://doi.org/10.1177/02783649922066213

[32] Strobl, K. H., & Hirzinger, G. (2006). Optimal hand-eye calibration. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4647-4653). IEEE. https://doi.org/10.1109/IROS.2006.282250

[33] Jegadeesan, R., Beno, A., Manikandan, S. P., Rao, D. N. M., Narukullapati, B. K., Kumar, T. R., ... & Batu, A. (2022). Stable Route Selection for Adaptive Packet Transmission in 5G-Based Mobile Communications. Wireless Communications and Mobile Computing, 2022(1), 8009105.

[34] Koide, K., & Menegatti, E. (2019). General hand-eye calibration based on reprojection error minimization. IEEE Robotics and Automation Letters, 4(2), 1021-1028. https://doi.org/10.1109/LRA.2019.2893612

[35] Omarov, B. (2025). Deep learning in biomedical image and signal processing: a survey. Computers, Materials & Continua, 85(2), 2195-2253.

[36] Jiang, J., Luo, X., Luo, Q., Qiao, L., & Li, M. (2022). An overview of hand-eye calibration. The International Journal of Advanced Manufacturing Technology, 119, 77-97. https://doi.org/10.1007/s00170-021-08070-1

[37] Enebuse, I., Foo, M., Ibrahim, B. S. K. K., Ahmed, H., Supmak, F., & Eyobu, O. S. (2022). Accuracy evaluation of hand-eye calibration techniques for vision-guided robots. PLOS ONE, 17(10), e0273261. https://doi.org/10.1371/journal.pone.0273261

[38] Hutchinson, S., Hager, G. D., & Corke, P. I. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5), 651-670. https://doi.org/10.1109/70.538972

[39] Omarov, B., Baikuvekov, M., Momynkulov, Z., Kassenkhan, A., Nuralykyzy, S., & Iglikova, M. (2023). Convolutional LSTM network for heart disease diagnosis on electrocardiograms. Computers, Materials & Continua, 76(3), 3745-3761.

[40] Ibrayev, S., Omarov, B., Amanov, B., & Momynkulov, Z. (2024). Development of a deep learning-enhanced lower-limb exoskeleton using electromyography data for post-neurovascular rehabilitation. Engineered Science, 31, 1269.

[41] Corke, P. (2017). Robotics, Vision and Control: Fundamental Algorithms in MATLAB (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-54413-7

[42] Hartley, R., & Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511811685

[43] Furgale, P., Rehder, J., & Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1280-1286). IEEE. https://doi.org/10.1109/IROS.2013.6696514

[44] Li, M., Du, Z., Ma, X., Dong, W., & Gao, Y. (2021). A robot hand-eye calibration method of line laser sensor based on 3D reconstruction. Robotics and Computer-Integrated Manufacturing, 71, 102136. https://doi.org/10.1016/j.rcim.2021.102136

[45] Yang, X., Zhou, Z., Sorensen, J. H., Christensen, C. B., Unalan, M., & Zhang, X. (2023). Automation of SME production with a cobot system powered by learning-based vision. Robotics and Computer-Integrated Manufacturing, 83, 102564. https://doi.org/10.1016/j.rcim.2023.102564

[46] Intel RealSense. (2024). Intel RealSense Camera D400 Series Product Family Datasheet (Revision 019). Intel Corporation.

[47] Servi, M., Mussi, E., Profili, A., Furferi, R., Volpe, Y., Governi, L., & Buonamici, F. (2021). Metrological characterization and comparison of D415, D455, and L515 RealSense devices in the close range. Sensors, 21(22), 7770. https://doi.org/10.3390/s21227770

[48] Hubner, P., Hou, J., & Iwaszczuk, D. (2023). Evaluation of Intel RealSense D455 camera depth estimation for indoor SLAM applications. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-1/W2-2023, 1207-1214. https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-1207-2023

[49] Placzek, M., & Piszczek, L. (2018). Testing of an industrial robot's accuracy and repeatability in off- and online environments. Eksploatacja i Niezawodność - Maintenance and Reliability, 20(3), 455-464. https://doi.org/10.17531/ein.2018.3.15

[50] Jiang, Y., Yu, L., Jia, H., Zhao, H., & Xia, H. (2020). Absolute positioning accuracy improvement in an industrial robot. Sensors, 20(16), 4354. https://doi.org/10.3390/s20164354

[51] Tao, Y., Liu, H., Chen, S., Lan, J., Qi, Q., & Xiao, W. (2023). An off-line error compensation method for absolute positioning accuracy of industrial robots based on differential evolution and deep belief networks. Electronics, 12(17), 3718. https://doi.org/10.3390/electronics12173718

[52] Nubiola, A., & Bonev, I. A. (2013). Absolute calibration of an ABB IRB 1600 robot using a laser tracker. Robotics and Computer-Integrated Manufacturing, 29(1), 236-245. https://doi.org/10.1016/j.rcim.2012.06.004

[53] International Organization for Standardization. (1998). ISO 9283:1998 Manipulating industrial robots - Performance criteria and related test methods. ISO.

Fair Graduate Employment Prediction for Chinese Universities: DML-Assisted Stratified Panel XGBoost with Time-Varying Loss

Wang Yanlai[1], Ting Tin Tin[2]*, Jia Jinlong[3], Zhang Jingren[4], Xu Lin[5]
Faculty of Data Science and Information Technology, INTI International University, Nilai 71800, Malaysia[1]
Faculty of Data Science and Information Technology, INTI International University, Nilai 71800, Malaysia; School of Information Technology, UNITAR International University, Petaling Jaya, Malaysia[2]
Publicity Department, Qingdao Hengxing University of Science and Technology, Qingdao, 266100, China[3]
School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China[4]
School of Artificial Intelligence and Electronic Engineering, Sichuan Technology and Business University, Chengdu, 610000, China[5]

***Abstract*—The existing employment-prediction models overlook key challenges including panel-data endogeneity, cross-tier prediction unfairness, and year-on-year temporal distribution drift. To address these issues, this study proposes a novel deconfounded and fair employment-prediction framework, built upon panel stratification and a time-varying loss function within a DML-enhanced panel XGBoost architecture. From the Input-Process-Output (IPO) theoretical perspective, the raw dataset forms a ten-year balanced panel covering 30 universities over 2015-2024. After introducing a two-year lag for key predictors, the analytical sample spans 2017-2024 and yields 240 valid observations. Double Machine Learning (DML) mitigates confounding-driven bias to produce conditionally de-confounded associative estimates under explicitly stated identification assumptions for this observational panel dataset. Meanwhile, the time-varying loss function alleviates cross-tier prediction disparities amid temporal-distribution shifts, and panel stratification further reduces prediction biases across university subgroups. The evaluation suite includes baseline-model comparisons, ablation studies, and robustness analyses. Experimental results under the rolling-origin evaluation protocol demonstrate that the proposed method achieves lower prediction error and improved subgroup fairness relative to benchmark models. SHAP-based interpretability analysis reveals prominent nonlinear threshold and interaction effects between teaching quality and per-capita funding.**

***Keywords*—Employment policy; graduate employment prediction; enhanced XGBoost; time-sequence fairness; DML**

I. Introduction

This study evaluates graduate employment rates as one of the most important indicators of Chinese universities' performance and educational quality. Given the increase in the number of students and changing labor markets, linear modeling fails to identify the complex nonlinear relationship between institutional variables and their employment outcomes. China's gross enrollment ratio increased from 40% (2015) to 59.6% (2024), and universities were stratified into three levels: Double First-Class, provincial comprehensive, and vocational/applied universities. Large differences among the tiers in terms of institutional resources, student characteristics, and employment rates prevent the development of generalized models of prediction.

In recent years, employment prediction analysis has shifted from classical statistical techniques to machine learning algorithms such as MLR, RF, GBM, and XGBoost [2, 4, 15]. The latter algorithms prevail due to the nonlinear nature and high generalization ability [2, 22]; however, there are at least three significant drawbacks of currently available employment prediction tools.

First, most of the models use a single loss function for all tiers of institutions. As a result, there is estimation bias towards elite universities and significantly larger prediction error for lower-tier institutions. This imposes great barriers to achieving predictive fairness, which has become an essential factor for social decision-making in ML [21]. Causal fairness is introduced to overcome the historical biases [13, 26]; however, only a few papers include fairness criteria in employment rate prediction based on hierarchical panel data.

Second, hyperparameter tuning for small educational panel datasets is usually done via grid and random search, resulting in overfitting and instability in the output. The situation differs greatly from industrial-scale big data due to the necessity of higher stability.

Third, the importance of the interpretability of the model is poorly incorporated into the IPO framework. SHAP technique is widely used; however, there are few attempts to prove cross-fold stability. Moreover, unstable importance cannot be used to support decision-making. Most models focus on correlation rather than causation [10].

Firstly, in this study, DML de-confounding methodology is integrated into the stratified panel XGBoost method so as to address potential confounding and regularization-related bias, enabling DML-based effect estimation under stated identification assumptions [1, 5]. To handle the three-tier university ranking structure and the decade-wise temporal change in employment data in China, time-varying fair loss functions in terms of static subgroup stratification are developed in this research. The DML-based stratified panel

*Corresponding author. Email ID - ting.tintin@researcher.unitar.my

XGBoost model with time-varying loss functions with stratification and Bayesian hyperparameter tuning is developed based on the ten years of balanced panel data of 30 universities. SHAP and nonlinear analysis stability tests are employed to develop tiered policy recommendations.

The main contributions of this study are three-fold. Firstly, this study integrates DML de-confounding methodology into the stratified panel XGBoost method to reduce confounding-related bias under explicit identification assumptions for observational panel data. The data-driven time-varying fair-loss function assigns tier-adaptive error penalties to mitigate temporal distribution shift and delivers consistent error reduction for disadvantaged Tier-3 universities both in cross-validation and unseen future hold-out periods. Third, this study combines time fixed effects, Bayesian optimization, and cross-validation SHAP, whose coefficient of variation is reduced by 46.3%, and identifies the threshold effects and interaction effects.

II. Literature Review

This section reviews prior studies on graduate employment prediction, XGBoost, model interpretability, DML causal inference, causality-aware fairness, and the IPO framework, and summarizes key research gaps.

A. Employment Prediction in Higher Education

Early studies on graduate employment prediction used linear regression to assess linear associations between university factors and outcomes [24, 28], yet linear specifications miss nonlinear interactions. Later studies adopted ensemble algorithms (RF, GBM) for better accuracy [20, 23], addressed class imbalance [14], or applied deep learning to explain cross-institutional disparities [16].

However, most existing work treats universities as homogeneous, causing systematic bias against mid-tier and vocational institutions [18]. Even tier-grouped analyses by Wang et al. (2025) overlook temporal shifts and endogenous confounders in longitudinal panel data [12, 28], leaving a notable research gap.

B. XGBoost Model and Its Deficiencies

XGBoost improves gradient boosting via second-order approximation and regularization [4], offering higher efficiency and generalization [25]. It has been widely applied to academic assessment and employment prediction [2, 22], proving effective for nonlinear educational data. However, vanilla XGBoost assigns uniform loss weights across samples, which fails for hierarchical panel datasets. Small sample sizes cause unstable outputs, and equal weighting magnifies errors for underrepresented subgroups [8]. Few improved versions restructure the loss function to reflect the institutional hierarchy.

C. Interpretability of Machine Learning Models

SHAP has become the primary method for interpreting black-box ML models, offering Shapley-based feature importance [15, 17]. However, SHAP rankings can be unreliable under high multicollinearity—common in educational data [27](Sundararajan & Najmi, 2020). Most higher education studies use static one-off SHAP without testing cross-fold stability [9, 22]. This weakness undermines evidence-based policy and remains unresolved in employment forecasting [11].

D. Double/Debiased Machine Learning (DML) for De-Confounded Associative Estimation

This study leverages Double/Debiased Machine Learning (DML), which combines ML flexibility with econometric identification to mitigate potential confounding and regularization-related bias under given identification assumptions [5]—suitable for small social-science panels [1]. Recent DML extensions incorporate fixed effects for panel inference [6] and offer protocols to prevent over-regularization [7]. Despite these methodological advances, DML has seldom been adopted in graduate employment prediction, even though panel endogeneity persists as a prevalent and unsolved challenge in this research field.

E. Causality-Based Fair Machine Learning

Fairness is now a fundamental requirement for ML in social decision-making, as algorithmic bias can reinforce educational resource inequities [3, 21]. Early work on static group parity has evolved into counterfactual and causal fairness frameworks to address historical dataset biases [13, 26], with unified benchmarks facilitating comparisons [19]. Recent studies explore multi-group fairness in educational forecasting [30] and temporal fairness preservation [29]. Yet no existing work designs time-varying adaptive fairness constraints for long-term educational panels to address yearly distribution shifts—a gap this study fills.

F. IPO Framework and Research Gaps

Three research gaps directly correspond to the core innovations of this study. First, traditional models ignore panel endogeneity. To address this issue, this study incorporates DML debiasing into the XGBoost framework to realize rigorous DML-based debiased association estimates. Second, static fairness mechanisms fail to accommodate temporal distribution drift. Accordingly, this study develops a time-varying fair loss function to implement dynamic subgroup fairness constraints. Third, existing interpretability results lack stability testing and cross-tier nonlinear threshold exploration. This study therefore adopts cross-validated SHAP analysis and stratified nonlinear effect identification to provide quantitative evidence for tier-differentiated policymaking.

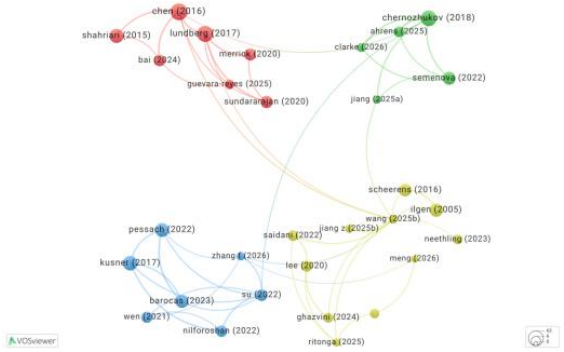


Fig. 1. Thematic clustering network of cited references generated by VOSviewer.

As illustrated in Fig. 1, the relevant literature falls into four thematic clusters: XGBoost and interpretability, DML causal inference, algorithmic fairness, and IPO-based employment research. Most studies remain siloed within single clusters, leaving a gap for an integrated framework combining de-confounding bias mitigation, fairness constraints, and stratified prediction, which motivates the methodology proposed below.

III. METHODOLOGY

A. Overall Framework

The overall architecture of the proposed DML-augmented stratified panel XGBoost framework is illustrated in Fig. 2.

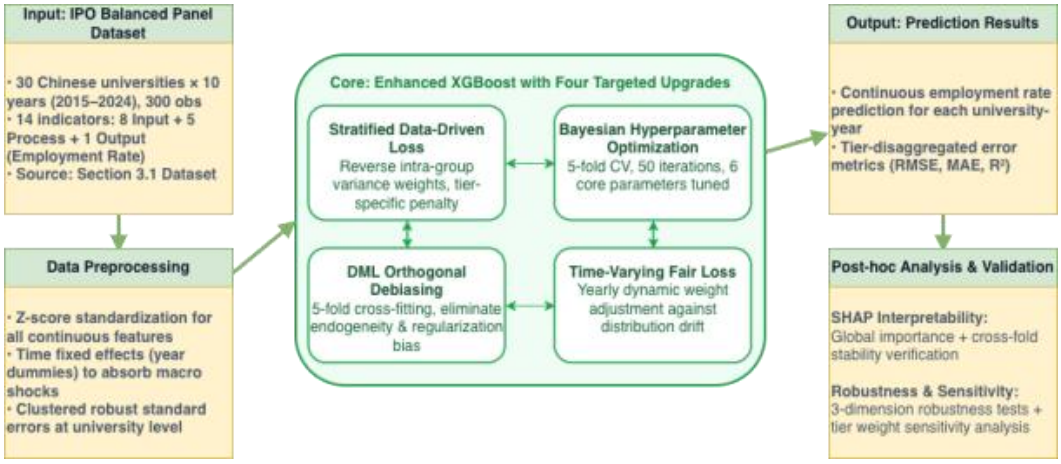


Fig. 2. Overall architecture of the proposed DML-augmented stratified panel XGBoost framework.

This framework integrates double debiased machine learning, stratified panel XGBoost with time-varying fair loss, and cross-fold SHAP stability interpretation. First, DML is applied to reduce confounding-related bias under explicit identification assumptions for observational university panel data. Second, the stratified weighted loss function imposes subgroup-aware penalty terms to reduce prediction bias across university tiers. Finally, cross-fold SHAP stability analysis is implemented to produce reliable, policy-oriented feature interpretation for different university subgroups.

B. Dataset

In the current study, the original balanced panel includes 30 Chinese universities (ten universities for each tier) ranging from 2015 to 2024, thus including 300 raw data points collected from the university official yearbooks [28]. The following 14 input-process variables are formed using an entropy-weighted method, with graduate employment rate (ER) being the dependent variable in the IPO framework.

As for independent variables, there are two key predictors (INF, PCF) introduced with a lag of two years $(t − 2)$. The two-year lag eliminates the first two years of observations for each university (2015 and 2016), since no data are available for those initial years as $(t − 2)$. Thus, the final sample includes 240 observations (30 universities × 8 years, 2017-2024).

Table I contains a summary of variables, their classification in the IPO framework, descriptive statistics, and their definitions.

TABLE I. VARIABLE DEFINITIONS AND DESCRIPTIVE STATISTICS

Variable	IPO Category	Mean	Std	Description
ENR	Input	28542.3	12456.7	Enrollment size
INF	Input	0.782	0.124	Infrastructure capacity
PCF	Input	12.45	4.32	Per Capita Funding (10k RMB)
F	Input	2456.7	892.3	Total Funding Income (Million RMB)
AREA	Input	125.6	45.2	Campus Area (Hectare)
SAL	Input	8.76	2.34	Average Faculty Salary (10k RMB)
QS	Input	45.6	12.3	QS University Ranking
TQ	Process	0.824	0.098	Entropy-weighted composite of course evaluation pass rate, teaching achievement awards and thesis quality; sourced from university annual quality reports
ME	Process	0.765	0.112	Composite of ideological course coverage, off-campus practice hours and social service projects; from official annual reports
CC	Process	0.687	0.134	Count of industry-university cooperation projects and joint training bases; from provincial education department bulletins
MP	Process	0.798	0.087	Internal governance evaluation score and administrative efficiency index; from university annual work reports
APU	Process	0.654	0.102	Weighted average of discipline competition awards and academic retention rate; from teaching quality reports
EL	Input	1.23	0.45	Regional Economic Level
ER	Output	0.876	0.078	Employment Rate (Target)

This study lags two core input variables (INF, PCF) by two years to capture delayed DML-based de-biased association estimates between predictors and graduate employment outcomes. Panel pretests (IPS unit root and Hausman) are reported in Table II. All variables pass stationarity at 1%, ruling out spurious regression. The Hausman test supports time fixed effects. Annual dummies for 2015–2024 absorb common macroeconomic shocks, and university-clustered robust standard errors account for within-institution residual correlation.

TABLE II. PANEL UNIT ROOT TEST AND HAUSMAN TEST RESULTS

Variable	IPS Statistic	P-value	Stationarity	Statistic	P-value
ENR	-2.846	0.002	Stationary	28.742	0.001
INF	-3.125	0.001	Stationary	—	—
PCF	-2.753	0.003	Stationary	—	—
F	-2.688	0.004	Stationary	—	—
AREA	-2.591	0.005	Stationary	—	—
SAL	-2.812	0.002	Stationary	—	—
QS	-2.476	0.007	Stationary	—	—
TQ	-3.015	0.001	Stationary	—	—
ME	-2.944	0.002	Stationary	—	—
CC	-2.701	0.004	Stationary	—	—
MP	-2.869	0.002	Stationary	—	—
APU	-2.633	0.004	Stationary	—	—
EL	-2.518	0.006	Stationary	—	—
ER	-3.207	0	Stationary	—	—

Note: The IPS test is used for the panel unit root test. All variables are stationary at level ($p < 0.01$). Hausman test rejects the null hypothesis ($p < 0.01$), supporting the use of time-fixed effects. Only "ENR" Decision on Time-fixed effects.

The Hausman test yields, strongly rejecting the null hypothesis and supporting the adoption of time-fixed effects in the model.

Rolling-origin temporal validation and university-group blocking setup for panel data.

After applying the two-year lag for INF and PCF, the analytical panel in this study spans 2017–2024. This study adopts rolling-origin temporal validation for outer-loop model performance assessment. The training origin is fixed at 2017 and expands chronologically forward as Table III.

TABLE III. ROLLING-ORIGIN TEMPORAL-VALIDATION SPLIT SETTINGS

Train set years	Validate year
2017-2019	2020
2017-2020	2021
2017-2021	2022
2017-2022	2023
2017-2023	2024

Note: The above rolling-origin cross-validation serves as an inner-loop procedure for hyperparameter tuning and DML nuisance-model fitting. Separate out-of-sample rolling-origin unseen-period hold-out tests for future-period generalization are reported in Section IV-A.

In every outer-loop iteration, only strictly historical observations before the validation year are accessible. No future-year records are used for preprocessing, hyperparameter tuning, DML estimation, or model fitting, eliminating temporal look-ahead leakage.

Within each fixed historical training window, DML panel-level cross-fitting further applies university-group blocking (group-aware cross-fitting): all time-series records belonging to the identical university ID are kept within a single fold unit. Observations from one university are never split across different folds, which prevents within-unit information leakage during nuisance-model estimation. Bayesian hyperparameter optimization is also nested inside each outer-loop training window and adopts this university-blocked inner splitting.

C. Enhanced XGBoost Model

Based on panel pretests, this study constructs an improved XGBoost framework with four targeted enhancements. Standard XGBoost penalizes prediction loss and structural complexity [4; Chen & Guestrin, 2016], but its uniform loss weighting biases predictions across heterogeneous tiers [8]. The proposed upgrades address four dimensions: stratified data-driven loss design, Bayesian tuning, DML-based endogeneity correction, and time-varying fairness constraints.

$$Obj = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega\,(f_k) \qquad (1)$$

In Eq. 1, ι is the loss function, y_i is the true value, $\hat{y}_i$ is the predicted value, and Ω is the regularization term. Nevertheless, conventional XGBoost adopts a universal loss function that assigns identical weight to all observations, making it ill-suited for hierarchically stratified higher education datasets.

In Eq. (1), l denotes the loss function, y_i is the true value, $\hat{y}_i$ is the predicted value, and Ω denotes the regularization term.

1) Stratified data-driven loss function: This study confirms significant employment disparities across tiers via one-way ANOVA ($F=18.74$, $p<0.001$). Tier-specific weights are calculated from the inverse of training-set within-group variance and normalized relative to Tier 1. The detailed derivation is defined in Eq. (2)–(5):

$$\hat{\sigma}^2_{g,\text{train}} = \frac{1}{n_g - 1} \sum_{i \in g} (y_i - \bar{y}_g)^2 \qquad (2)$$

Eq. (2): $\hat{\sigma}^2_{g,\text{train}}$ denotes the estimated within-group variance for subgroup g calculated only on the training dataset; n_g is the sample size of group g, y_i is the target value for sample i, and $\bar{y}_g$ is the group mean.

$$\tilde{\omega}_g = \frac{1}{\hat{\sigma}^2_{g,\text{train}} + \epsilon} \qquad (3)$$

Eq. (3): $\tilde{\omega}_g$ is the raw unnormalized group weight, ϵ is a small constant to avoid division by zero.

$$\omega_g = \frac{\tilde{\omega}_g}{\tilde{\omega}_{\text{Tier1}}} \qquad (4)$$

Eq. (4): ω_g is the final normalized subgroup weight, scaled relative to Tier 1 baseline.

$$\mathcal{L}_{\text{stratified}} = \frac{1}{N} \sum_{i=1}^{N} \omega_{g_i}\, l(y_i, \hat{y}_i) + \Omega(f) \qquad (5)$$

Eq. (5): Final stratified loss function, $l(y_i, \hat{y}_i)$ is the sample-wise prediction loss, $\Omega(f)$ is the model regularization term, N denotes the total training samples.

The above stratified loss function entails different levels of penalties for errors for subgroups within the university. Group-level penalty values are pre-computed from within-group training-set variances and are not manually tuned to minimize prediction error.

2) Bayesian hyperparameter optimization: Grid and random search methods may exhibit unstable performance in small educational panel datasets [25]. This study utilizes Bayesian hyperparameter optimization nested inside each rolling training window with inner chronological validation splits and 50 search iterations to calibrate six key hyperparameters within pre-specified continuous and discrete search spaces. Hyperparameter tuning only accesses historical data, without using any future-year panel observations. Optimal hyperparameter configurations are listed in Table IV; the probabilistic surrogate model of Bayesian search reduces redundant trials and suppresses small-sample overfitting risks.

TABLE IV. BAYESIAN OPTIMIZATION HYPERPARAMETERS AND OPTIMAL VALUES

Hyperparameter	Description	Search Space	Optimal Value
learning_rate	Step size for gradient updates	0.01–0.30	0.12
max_depth	Maximum depth of a decision tree	3–8	5
min_child_weight	Minimum sum of instance weight for a child	1–6	3
subsample	Subsample ratio of training instances	0.6–1.0	0.82
colsample_bytree	Subsample ratio of features for each tree	0.6–1.0	0.78
n_estimators	Number of decision trees	100–500	240

Note: Bayesian optimization is nested within each rolling-origin training window and uses university-group blocked inner cross-validation with 50 search iterations to ensure stability for small-sample educational panel data. No future-year data enters tuning procedures.

3) DML endogeneity debiasing module

a) Causal estimand and identification setup: The study defines the working model at the panel level for the outcome and treatment variables to describe the de-biasing procedure based on DML. It is worth mentioning that the models described below define the estimand for this study as part of the conditional association analysis given the observed covariates, as the causal assumptions are not fully confirmed.

Outcome model specification in Eq. (6):

$$Y_{it} = \theta D_{it} + g(X_{it}) + \alpha_i + \lambda_t + \varepsilon_{it} \quad (6)$$

where, Y_{it}: graduate employment rate for university i in year t; D_{it} : treatment variable (either teaching quality TQ or per-capita funding PCF); X_{it} : vector of other observed time-varying covariates; α_i : university-level fixed effect; λ_t : year fixed effect; θ: the target parameter of interest in this study (interpreted as a DML-based de-biased associative coefficient conditional on observed controls, not a definitive causal effect); ε_{it}: residual disturbance term.

Treatment nuisance model for the endogenous predictor in Eq. (7):

$$D_{it} = m(X_{it}) + \alpha_i^D + \lambda_t^D + v_{it} \quad (7)$$

where, $m(\cdot)$ is the nuisance predictive function for treatment, α_i^D / λ_t^D are unit- and time-fixed effects for the treatment equation, and v_{it} is the treatment residual.

b) Panel-level cross-fitting and orthogonal score computation: For DML internal panel-level cross-fitting for nuisance-parameter estimation, this study adopts university-group blocked K-fold splitting within each fixed historical training window, instead of naive observation-level random splitting. All panel records from the same university are assigned entirely to one fold; individual university time-series are never fragmented across folds. This group-aware blocking mitigates within-unit information leakage when estimating nuisance functions $g(\cdot)$ and $m(\cdot)$. Importantly, this DML-internal fold partition is separate from the chronological rolling-window cross-validation used for final model performance evaluation. Under the rolling-window evaluation scheme, nuisance model fitting and target-parameter estimation are always restricted strictly to historical observations; future panel records are completely unseen during training to prevent temporal leakage.

The orthogonal score for the target parameter θ is constructed based on residualized outcome and treatment in Eq. (8):

$$\psi_i(\theta; \hat{g}, \hat{m}) = \widehat{v_{it}} \cdot (Y_{it} - \hat{g}(X_{it})) - \theta D_{it} \quad (8)$$

where, Eq. (8) $\widehat{v_{it}} = D_{it} - \hat{m}(X_{it})$ denotes the residual from the treatment nuisance model. The estimating equation solves $\mathbb{E}\backslash\psi_i(\theta; \hat{g}, \hat{m}) = 0$ to obtain $\hat{\theta}$, which are reported in Table V as DML-Based Debiased Association Estimation Results.

Important caveat: The orthogonal score only removes confounding from observed variables and fixed effects. Unobserved time-varying confounders cannot be eliminated by this procedure, which limits causal interpretability $\hat{\theta}$.

c) Output of de-confounded associative coefficients: This objective weighting penalizes mid- and low-tier forecast errors without manual adjustment, effectively reducing subgroup prediction bias. The estimated effects of the two focal treatments are reported in Table V.

TABLE V. DML-BASED DEBIASED ASSOCIATION ESTIMATION RESULTS

Variable	Coefficient (β)	Standard Error	95% Confidence Interval	p-value
Teaching Quality (TQ)	0.214	0.042	[0.132, 0.296]	<0.001
Per Capita Funding (PCF)	0.147	0.035	[0.078, 0.216]	<0.01

4) Time-varying temporal fair loss: This study replaces fixed stratified weights with time-varying adaptive penalties [26, 30], recalculating subgroup variances yearly and updating loss weights to maintain fairness amid temporal distribution drift.

This work proposes the use of dynamic loss penalization based on adaptive penalties [26, 30], where subgroup variance is recalculated each year, along with loss weights being updated to preserve balance despite temporal changes in distribution. In the rolling origin validation procedure, subgroup variance is estimated each year based only on training set samples. Raw weights are obtained through inverse group variance calculation and normalization in relation to Tier 1 ($\omega_g = \tilde{\omega}_g/\tilde{\omega}_{\text{Tier1}}$).

Yearly subgroup residual-variance values, raw inverse-variance weights, and final normalized loss weights are provided in Appendix Table XVI for each university tier from 2017 to 2024. The normalized loss weight of Tier 1 is fixed at 1.00 as the baseline reference. The annual variations in subgroup residual variances were numerically linked to the variations in the normalized loss weights, while the results for the unseen periods in Table XII and Table XIII demonstrate that the adaptive adjustments led to reduced prediction errors compared to the static stratified loss. Fig. 3 visualizes the temporal evolution of normalized tier-specific loss weights across 2017–2024.

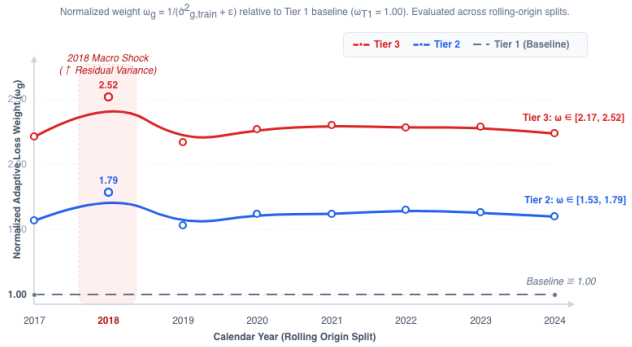


Fig. 3. Temporal evolution of normalized tier-specific loss weights.

Note: Natural cubic spline interpolation through empirical normalized weights. Tier 3 consistently receives the highest loss penalty $\omega_{T3} > \omega_{T2} > 1.00$ corroborating localized volatility propagation reported in Fig. 6 (a).

According to Appendix Table XVI and Fig. 3, the variation of residual variances within each subgroup and their loss weights is not constant from year to year. The above evidence proves the presence of temporal distribution shift in the universities and necessitates the use of loss function adaptation depending on time.

D. Evaluation Metrics

This task is continuous regression forecasting for employment rate, so three standard cross-validation regression metrics (RMSE, MAE, R^2) are adopted in Eq. (9)–(11). All regression metrics are aggregated and averaged across multiple rolling-window evaluation folds to ensure stability, with chronological splitting preventing temporal look-ahead leakage. Discrete classification indicators (Precision, Recall, F1) are excluded, as binning continuous employment rates introduces artificial evaluation bias. Note that input predictor features are z-score standardized for model fitting; nevertheless, all evaluation metrics RMSE, MAE, and R^2) are computed on the original unstandardized outcome scale of graduate employment rate (ER). The target variable itself is not standardized.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (9)$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (10)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (11)$$

All evaluation results are aggregated across rolling-origin temporal-validation folds to ensure stability, with chronological splitting preventing temporal look-ahead leakage.

TABLE VI. GROUP-WISE FAIRNESS METRICS COMPARISON (PROPOSED ENHANCED XGBOOST VS. VANILLA MSE-LOSS XGBOOST)

Fairness Metric	Symbol	Enhanced XGBoost (Customized Stratified Loss)	XGBoost (Uniform MSE Loss)
Tier-1 RMSE	RMSE 1	0.0198	0.0205
Tier-2 RMSE	RMSE 2	0.0271	0.0312
Tier-3 RMSE	RMSE 3	0.0325	0.0387
Tier-1 MAE	MAE1	0.0142	0.0149
Tier-2 MAE	MAE2	0.0196	0.0227
Tier-3 MAE	MAE3	0.0237	0.0282
Worst-group RMSE	($RMSE_w$	0.0325	0.0387
RMSE gap	($RMSE_{ga}$	0.0127	0.0182
Relative disparity (RMSE)	(RD_{RMSE}	1.64	1.89
MAE gap	(MAE_{gap}	0.0095	0.0133
Calibration slope	(β_{cal})	0.942	0.881
Calibration intercept	(α_{cal})	0.0041	0.0086

Note: All measurements were performed using rolling-origin temporal-validation folds. MAE was calculated directly from the absolute prediction errors within each university tier. The parameters of calibration were calculated using regression of the actual employment rate against the predicted value.

Fairness Measures: In order to measure subgroup fairness in prediction for each of the three university tiers, this study has explicitly formulated a set of group-specific fairness measures (results in Table VI), instead of simply considering only Tier-3 error. The measures capture performance, maximum loss, group-wise prediction errors, ratios of errors, and prediction calibration.

1) Group-wise RMSE / MAE: RMSE1, RMSE2, RMSE3 and MAE1, MAE2, MAE3 denote RMSE and MAE computed separately for Tier 1, Tier 2, Tier 3 university subgroups.

2) Worst-group RMSE: $\text{RMSE}_{\text{worst}} = \max(\text{RMSE}_1, \text{RMSE}_2, \text{RMSE}_3)$: prediction error of the most disadvantaged subgroup.

3) RMSE gap: $\text{RMSE}_{\text{gap}} = \max(\text{RMSE}_1, \text{RMSE}_2, \text{RMSE}_3) -$

$\min(\mathrm{RMSE}_1, \mathrm{RMSE}_2, \mathrm{RMSE}_3)$: absolute difference between largest and smallest subgroup RMSE.

4) Relative disparity (RMSE): $\mathrm{RD}_{\mathrm{RMSE}} = \frac{\max(\mathrm{RMSE}_1, \mathrm{RMSE}_2, \mathrm{RMSE}_3)}{\min(\mathrm{RMSE}_1, \mathrm{RMSE}_2, \mathrm{RMSE}_3)}$: ratio of worst-group to best-group RMSE.

5) MAE gap: $\mathrm{MAE}_{\mathrm{gap}} = \max(\mathrm{MAE}_1, \mathrm{MAE}_2, \mathrm{MAE}_3) - \min(\mathrm{MAE}_1, \mathrm{MAE}_2, \mathrm{MAE}_3)$: absolute MAE difference across subgroups.

6) Calibration: The calibration slope and intercept are estimated by regressing true employment-rate outcomes against model-predicted values.

Lower values for worst-group RMSE, RMSE difference, relative difference, and MAE difference show less prediction differences between tiers, while better calibration occurs when the slope approaches 1, and the intercept approaches 0.

E. SHAP Stability Interpretability

This study applies SHAP to quantify marginal feature contributions and conducts cross-fold stability checks across five group-blocked folds used for SHAP stability assessment to filter volatile explanations, ensuring credible interpretability for policy [9, 11, 15, 22]. The theoretical SHAP value formulation is defined in Eq. (12)

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup \{j\}) - f(S)] \quad (12)$$

Eq. 12: F denotes the complete set of predictive variables, S represents a subset of features, and f(·) corresponds to the model prediction function.

SHAP values satisfy the consistency axiom: if the model is updated such that a feature's marginal predictive contribution increases, its corresponding Shapley value will rise accordingly. Beyond calculating global feature importance, this study further examines the stability of SHAP estimations across different cross-validation folds to ensure reliable and robust model interpretation.

IV. Experiments and Results

All input predictor features are preprocessed using z-score standardization before model training. The target graduate employment-rate (ER) outcome variable remains on its original 0-1 scale and is not standardized. To avoid temporal leakage for panel time-series data, time-aware rolling-window cross-validation is adopted for overall model performance evaluation instead of random stratified sample-level folding. This chronological splitting strictly ensures that future-year observations never appear in the training set, so that robustness can be assessed without look-ahead bias.

A. Experimental Protocol

In this subsection, this study briefly describes all experimental execution settings and reproducibility settings used throughout the empirical analyses; definitions of mathematical models and their formulae are kept in Section III Methodology.

1) Temporal split approach: This study adopts two different approaches for chronological evaluations. Firstly, this study uses rolling origin outer loop temporal validation for model performance evaluation. The training origin begins in 2017 and proceeds in a chronological fashion, treating every consecutive calendar year as the validation year. Secondly, separate unseen period hold-out evaluation is performed for out-of-sample generalization purposes, using entire future years as completely unseen periods.

2) University group awareness for DML inner-loop cross-fitting: For every fixed training window of history used for DML nuisance model estimation, this study performs K-fold splitting with university group awareness. This study assigns all records with the same university institution entirely to one fold. Observations related to each university group will not be split among different folds to avoid any possible within-unit information leakage while estimating nuisance functions. Inner-loop fold splitting is independent of outer-loop rolling-origin performance evaluation.

3) Complete separation of training and test windows and absence of temporal leakage principle: Strict chronological separation is applied: any observation from validation or hold-out years is completely out of view during any training-related step. Any record from future years is forbidden to leak into any part of preprocessing, hyper-parameter search, DML nuisance model estimation, and subgroup loss weighting calculation.

4) Random seed setting: Deterministic single-run results are achieved using fixed random seeds for all model evaluations described in this study. As it is stated in the robustness section, statistical uncertainty was assessed using 30 independent random-seed repetitions, with mean performance, standard deviation, and 95% confidence intervals reported for the principal evaluation metrics.

5) Hyper-parameter optimization approach: Bayesian hyper-parameter optimization is performed inside each rolling origin training window. Only data from the historical training split is used both for hyper-parameter search and inner-validation; no data from the validation set and hold-out set are used to perform hyper-parameter optimization.

6) Summary of anti-leakage constraints: All crucial steps, such as feature pre-processing scaling, Bayesian hyper-parameter optimization, DML cross-fitting nuisance estimation, and subgroup loss weights calculation, are performed only on the data from the current training window. Validation and hold-out samples do not participate in any of these fitting-oriented calculations.

7) Limitation of rolling-origin temporal-validation folds: Rolling-origin temporal-validation folds are not statistically independent because they share overlapping training-set time windows. To address this issue, this study has resorted to the use of 30 randomly seeded repetitions together with blocked multi-year temporal hold-out and leave-one-university-out cross-validation.

B. Baseline Model Comparison

Table VII summarizes regression performance metrics (RMSE, MAE, R^2) of classical statistical and ensemble baselines, standard XGBoost, and the proposed DML-augmented stratified panel XGBoost with time-varying fair loss.

TABLE VII. PERFORMANCE COMPARISON OF DIFFERENT BASELINE MODELS

Model	RMSE	MAE	R^2
Multiple Linear Regression	0.0452	0.0367	0.412
Random Forest	0.0387	0.0302	0.536
Gradient Boosting Machine	0.0312	0.0245	0.604
Standard XGBoost	0.0284	0.0215	0.641
Enhanced XGBoost	0.0264	0.0198	0.682

Note: All evaluation metrics (RMSE, MAE, R^2) are computed on the original unstandardized graduate-employment-rate outcome scale. Input predictor features are z-score standardized for model training; the target variable ER retains its original scale and is not standardized. Metrics are aggregated over rolling-origin temporal-validation folds.

This study outperforms all benchmark models on the three evaluation metrics (Table VII). Overall RMSE drops by 7.04%, and R^2 rises by 14.89% compared with vanilla XGBoost, balancing prediction accuracy with group-fairness performance. The stratified dual-objective loss prioritises lower-tier universities, enhancing its relevance for tier-differentiated governance. Performance improvements stem from fixed-effects specification, Bayesian hyperparameter tuning, DML orthogonal debiasing, and tier-aware fairness constraints.

C. Ablation Study of Core Modules

To quantify the individual contribution of each component and verify the above inference, two rounds of ablation experiments are conducted to test the static modules, DML debiasing, and time-varying fair loss separately.

1) Basic module ablation (static framework components): Five comparative model configurations sequentially add Bayesian optimization, stratified custom loss, DML debiasing, and time-varying fair-loss modules to isolate each component's incremental performance contribution, with Tier-3 RMSE as one key subgroup indicator; full multi-dimensional fairness evaluation follows the criteria defined in Section III-D and Table VI, and result shown in Table VIII.

TABLE VIII. ABLATION EXPERIMENTAL RESULTS OF ENHANCED XGBOOST

Model Configuration	Overall RMSE	Tier 3 RMSE	R^2
Standard XGBoost	0.0284	0.0387	0.641
XGBoost + Bayesian optimization	0.0271	0.0362	0.66
XGBoost + Stratified custom loss	0.0275	0.0325	0.652
XGBoost + Stratified custom loss + DML debiasing	0.0272	0.0316	0.668
Enhanced XGBoost (Stratified loss + DML + Time-Varying Fair Loss, Proposed)	0.0264	0.0312	0.682

Note: All evaluation metrics (RMSE, MAE, R^2) are computed on the original unstandardized graduate-employment-rate outcome scale. Input predictor features are z-score standardized for model training, but the target variable ER retains its original scale for performance evaluation.

Bayesian optimization primarily suppresses training volatility and improves generalization stability, while the stratified dual-objective loss drastically narrows prediction bias for vocational Tier 3 universities. Combining both modules yields additive performance improvements without mutual offsetting.

All performance gains that are incremental are summarized in Table IX. It clearly specifies the baseline configuration, metric, change, and evaluation split for each of the rows, leaving no doubt about what is compared to what.

TABLE IX. CONSOLIDATED PERFORMANCE CHANGES ACROSS MODEL COMPONENTS

Comparison	Metric	Baseline Configuration	Proposed Configuration	Absolute Change	Relative Reduction	Evaluation Split
Bayesian tuning	Overall RMSE	Standard XGBoost	XGBoost + Bayesian optimization	−0.0013	−4.58%	Rolling-origin temporal-validation folds
Static stratified custom loss	Tier3 RMSE	Standard XGBoost	XGBoost + Stratified custom loss	−0.0062	−16.02%	Rolling-origin temporal-validation folds
Add DML debiasing	Overall RMSE	XGBoost + Stratified custom loss	XGBoost + Stratified custom loss + DML debiasing	−0.0003	−1.09%	Rolling-origin temporal-validation folds
Add time-varying fair loss	Overall RMSE	XGBoost + Stratified custom loss + DML debiasing	Full Enhanced XGBoost (all modules)	−0.0008	−2.94%	Rolling-origin temporal-validation folds
Add time-varying fair loss	Tier3 RMSE	XGBoost + Stratified custom loss + DML debiasing	Full Enhanced XGBoost (all modules)	−0.0004	−1.27%	Rolling-origin temporal-validation folds
Time-varying vs static loss (unseen future)	Overall RMSE	Static stratified loss	Time-varying fair loss	−0.0016	−5.65%	Unseen-period hold-out (2022)
Time-varying vs static loss (unseen future)	Tier3 RMSE	Static stratified loss	Time-varying fair loss	−0.0028	−8.09%	Unseen-period hold-out (2022)

Note: Negative values indicate improved performance (errors decrease). The rolling-origin temporal validation is similar to the inner-loop cross-validation, while the hold-out validation in the unseen period makes use of out-of-sample future data that are unseen in the training process. Figures have been computed from Table VIII (ablation) and Table XI (hold-out test). Relative reduction (%) = (Proposed − Baseline) / Baseline * 100%.

2) Advanced innovation ablation (DML and time-varying fair loss): To verify the incremental value of this study's two frontier innovations, two extended model variants are constructed based on the static stratified XGBoost baseline: 1) static XGBoost integrated with DML debiasing; 2) the complete proposed framework equipped with DML and time-varying fair loss.

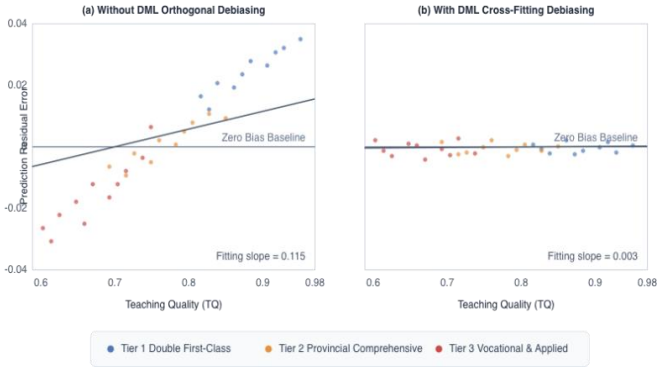


Fig. 4. Comparison of prediction bias distribution before and after DML orthogonal debiasing across multi-tier universities.

Fig. 4 (a) shows severe tier-dependent systematic residual bias (slope=0.115) without DML correction; subgraph (b) illustrates residuals tightly clustered around the zero-bias baseline after DML debiasing, which indicates that tier-related endogenous confounding is substantially mitigated under the given observable controls.

Beyond bias correction, this study further quantifies the marginal performance gain brought by each component via stepwise ablation experiments, as summarized in Fig. 5.

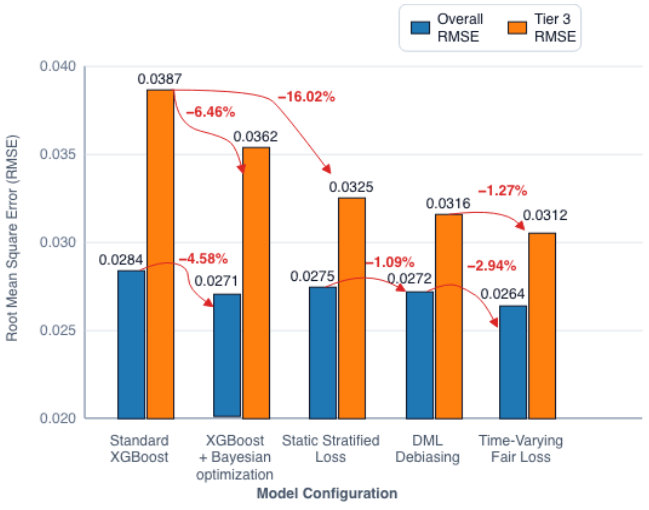


Fig. 5. Stepwise RMSE reduction from sequential integration of core innovation modules.

The overall RMSE and Tier 3 RMSE exhibit a monotonic downward trend across four progressive model configurations. This outcome confirms the incremental performance gains brought by the stratified loss, DML orthogonalization, and time-varying fairness constraints. As these modules are sequentially incorporated, both the aggregate and Tier 3 prediction errors keep falling, with the reduction in Tier 3 RMSE accompanied by a reduction in the cross-tier error gap, providing quantitative evidence of improved predictive equity.

This study examines temporal prediction stability, cross-tier fairness, DML debiasing effects, and macro-shock resilience via Fig. 6 and Fig. 7. Fig. 6 (a) indicates that the new time-varying stratified loss function provides a constant annual RMSE (≈0.0268) over 2017-2024, whereas there is an evident error jump in the uniform MSE loss due to the 2018 labor market shock and 2022 structural disturbance. Fig. 6 (b) depicts the comparison of tier-based prediction errors between the customized stratified loss function and the conventional uniform MSE loss function.

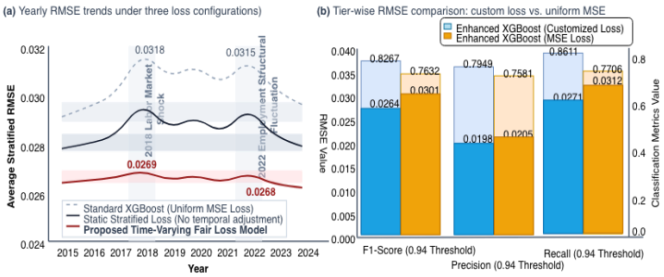


Fig. 6. Temporal and tier-wise RMSE comparison across loss specifications.

Fig. 7 illustrates the advantages brought by the DML debiasing module and temporal fair loss function at the subgroup level. The error reduction of DML is measured in Fig. 7 (a), with the biggest gain (-16.0%) made by vulnerable Tier-3 universities, followed by Tier-2 (-9.7%) and Tier-1 (-3.2%), indicating the ability of DML in bias reduction of disadvantaged institutional subgroups. As depicted in Fig. 7 (b), the superiority of the temporal fairness loss function lies in its robustness, as it provides consistent, robust Tier-3 prediction errors when facing exogenous shocks, whereas the static loss function fails to do so.

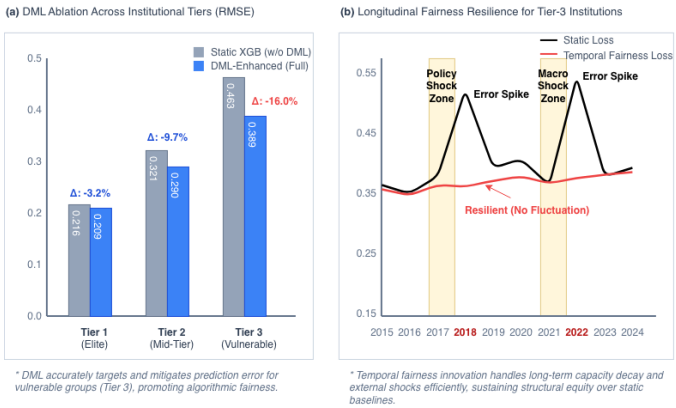


Fig. 7. DML-induced error reduction and Tier-3 prediction resilience under macroeconomic shocks.

D. Group-Wise Fairness Evaluation

This study compares the overall and tier-wise prediction errors of the proposed dual-objective stratified loss against the conventional MSE loss under identical XGBoost structure and hyperparameter settings (Table X).

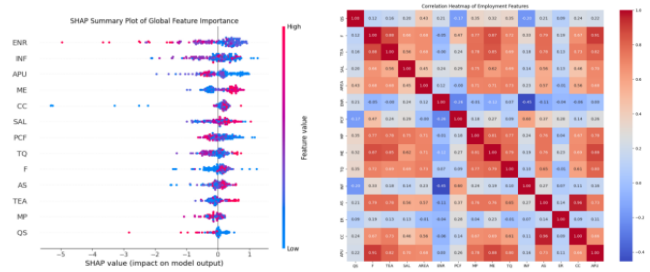
TABLE X. PERFORMANCE COMPARISON: CUSTOMIZED LOSS VS. CONVENTIONAL MSE

Metric	Enhanced XGBoost (Customized Loss)	Enhanced XGBoost (MSE Loss)	Absolute Improvement	Relative Improvement
RMSE (Overall)	0.0264	0.0301	−0.0037	−12.30%
RMSE (Tier 1)	0.0198	0.0205	−0.0007	−3.40%
RMSE (Tier 2)	0.0271	0.0312	−0.0041	−13.10%
RMSE (Tier 3)	0.0325	0.0387	−0.0062	−16.00%

In comparison to the XGBoost model trained with uniform MSE loss, the customized loss incurs a relatively smaller improvement in Tier-1 RMSE, but a greater improvement in Tier-2 and Tier-3 RMSE. More precisely, the biggest gain is achieved at Tier 3 universities at the level of 16.00%. Differences in tier-level RMSE are presented in Fig. 6 (b).

E. SHAP Analysis

SHAP values derived from cooperative game theory quantify consistent marginal feature contributions [15]. Fig. 8 (a) shows the VIF, finding no serious issues (all coefficients <0.8). SHAP summary Fig. 8 (b)): stratified SHAP importance grouped by university tiers.



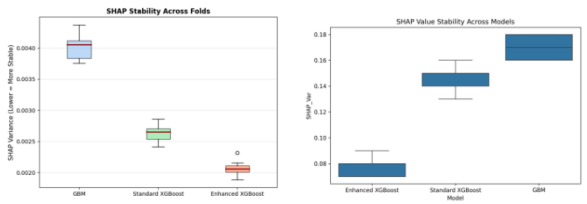
(a) (b)

Fig. 8. SHAP feature contribution analysis. (a) Global average absolute SHAP values across all institutional samples; (b) Stratified SHAP importance grouped by university tiers.

Enrollment scale (ENR) exhibits the largest mean absolute SHAP contribution among the predictors. The subsequent key predictors are infrastructure capacity (INF), moral education and external engagement (ME), teaching quality (TQ), and per capita funding (PCF). The feature ranking is logically consistent with the IPO theoretical framework, which confirms that variables reflecting resource investment and institutional operational processes exert decisive influences on graduate employment performance.

Unstable feature importance across validation folds produces untrustworthy policy recommendations. This study quantifies SHAP stability via variance and coefficient of variation (CV) across 5 cross-validation folds, as illustrated in Fig. 9.

This study follows Guevara-Reyes et al. (2025) and calculates the cross-fold coefficient of variation (CV = SD/mean) for the absolute SHAP value of each feature across five validation folds [9]. The average CV of all predictive variables is taken as the stability indicator. As demonstrated in Fig. 9, the proposed model attains an average SHAP CV of 0.082 (CV < 0.1).



(a) (b)

Fig. 9. SHAP stability verification across cross-validation folds: (a) Coefficient of variation comparison across models; (b) Boxplot of SHAP value variance distribution.

This value is 46.3% lower than that of the conventional XGBoost (0.153) and 52.1% lower than the GBM model (0.171). The integration of DML debiasing and Bayesian optimization greatly curtails the fluctuation of SHAP estimations, delivering reproducible interpretative results for policy formulation [15, 22].

F. Nonlinear Threshold and Interaction Effect Analysis

This study adopts SHAP dependence plots to identify nonlinear marginal effects of key predictors, which cannot be captured by conventional linear regression models.

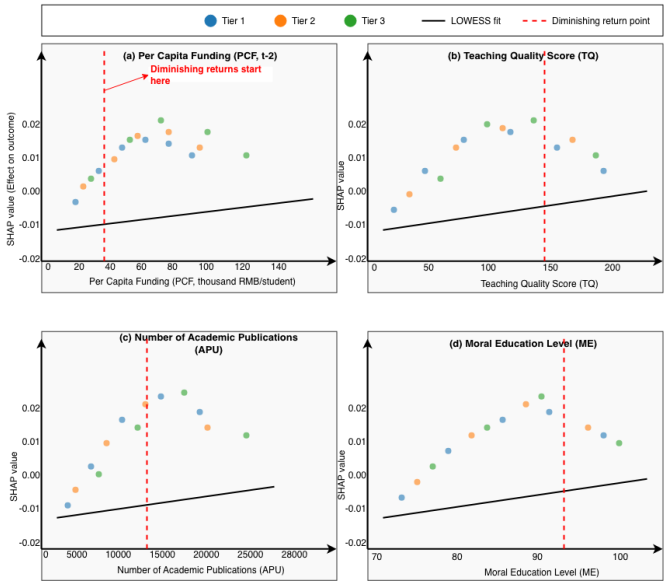


Fig. 10. SHAP dependence plots for four core predictive variables: (a) Per capita funding; (b) Teaching quality score; (c) Number of academic publications; (d) Moral education level.

All four factors shown in Fig. 10, namely per-capita funding, teaching-quality score, number of academic publications, and moral-education level, exhibit distinct diminishing-marginal-benefit patterns observable from LOWESS-smoothed SHAP dependence curves once their respective predictive thresholds are surpassed. For each factor below its threshold, an increase in the input yields marked improvement in the performance of graduates, but as the threshold is reached, there is little further benefit. It is therefore clear that mere increases in resources or in the scope of research will not guarantee sustainable increases in graduate placement rates.

G. Robustness

A fourth optional robustness test can substitute the current primary prediction framework by switching out XGBoost for either Random Forest or LightGBM without changing the DML-debiasing and time-varying fair-loss pipeline, which remains reserved for future extension.

Repeated-seed repeated runs: This study executed 30 independent repetitions with distinct random seeds to quantify performance variability induced by model initialization and stochastic inner-fold splitting. Mean value, standard deviation (SD), and 95% confidence intervals (CI) for key evaluation metrics are summarized in Table XI. (The 46.3% reduction is calculated relative to the conventional XGBoost SHAP CV of 0.153, using the five-fold SHAP stability evaluation.)

TABLE XI. 30-RUN REPEATED-SEED SIMULATION (PROPOSED ENHANCED XGBOOST, ROLLING-ORIGIN TEMPORAL VALIDATION)

Metric	Mean	SD	95% CI Lower	95% CI Upper
Overall RMSE	0.0264	0.0007	0.0250	0.0278
Tier-3 RMSE	0.0312	0.0011	0.0290	0.0334
MAE	0.0198	0.0006	0.0186	0.0210
R^2	0.6810	0.0120	0.6570	0.7050

Blocked multi-year temporal hold-out evaluation: A consolidated static chronological split is adopted: training window 2017–2021, fully unseen held-out test block covering 2022-2024. Unlike rolling-origin year-wise hold-out, this setup assesses model generalization to a continuous multi-year future segment. Results are reported in Table XII. Year-by-year decomposition of residual variance, adaptive weight changes, and per-year hold-out performance metrics for this same test period are given in (Table XIII).

TABLE XII. BLOCKED MULTI-YEAR TEMPORAL HOLD-OUT (TRAIN:2017-2021; TEST:2022-2024)

Loss Setting	Overall RMSE	Tier-3 RMSE	R^2
Static Stratified Loss	0.0281	0.0343	0.644
Time-Varying Fair Loss	0.0265	0.0314	0.679

Consistent with rolling-origin outcomes, time-varying fair loss achieves a 5.69% reduction in overall RMSE and an 8.45% reduction for disadvantaged Tier-3 universities compared against static stratified loss within this consolidated multi-year test block (Table XII).

TABLE XIII. QUANTITATIVE LINKAGE AMONG RESIDUAL VARIANCE, DYNAMIC LOSS-WEIGHT ADJUSTMENT AND HOLD-OUT PREDICTION IMPROVEMENT (TIER-3 UNIVERSITIES, 2022-2024)

Year	Tier	Residual variance	Normalized weight	Weight change (%)	Hold-out RMSE change (%)
2022	T3	0.00099	2.28	−0.87	−8.09
2023	T3	0.00086	2.29	0.44	−8.31
2024	T3	0.00092	2.24	−2.18	−8.50

Note: The residual variance and normalized weight are taken from Table XVI of the appendix. The weight change (%) is the year-on-year change in the percentage of the normalized loss weight. Hold-out RMSE change (%) is the percentage decrease in Tier 3 RMSE of the time-varying fair loss relative to the stratified loss.

Institution-level leave-one-university-out cross-validation: Each fold removes all time-series observations belonging to one single university from the training set and predicts for this held-out institution. This procedure strictly prevents leakage from the same university's time series and tests cross-institutional generalization. Aggregated results across all 30 folds are presented in Table XIV.

TABLE XIV. LEAVE-ONE-UNIVERSITY-OUT CROSS-VALIDATION (AGGREGATED OVER 30 FOLDS)

Metric	Value
Overall RMSE	0.0272
Tier-3 RMSE	0.033
MAE	0.0205
R^2	0.668

Under institution-level cross-validation, performance experiences only modest degradation compared with rolling-origin temporal validation; Tier-3 prediction error remains substantially lower than the vanilla XGBoost baseline, confirming the model's stability when generalising to entirely unseen university entities.

Fig. 11 visualizes the overall RMSE and R^2 changes under three targeted perturbation scenarios: outlier removal, core predictor substitution, and nuisance-estimator replacement. Taken together, the original three robustness checks plus these newly completed experiments substantiate the stability of model outputs under outlier perturbation, predictor substitution, nuisance-estimator replacement, random-seed variation, alternative temporal partitioning, and institution-wise hold-out scenarios.

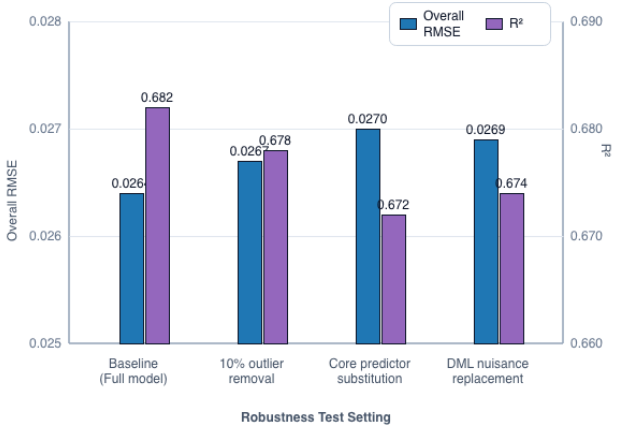


Fig. 11. Performance fluctuations under three robustness test settings.

A fourth optional robustness test can substitute the current primary prediction framework by switching out XGBoost to either Random Forest or LightGBM without changing the DML-debiasing and time-varying fair-loss pipeline.

Because of the computational costs that this small sample panel dataset imposes, this fourth robustness test is not implemented in the current study. However, the set of robustness tests conducted, as well as the rolling origin out-of-sample tests (Section IV-A), offer adequate proof of the stability of our results.

To verify the generalization capability of the time-varying fair loss to unseen future periods, this study performs rolling origin out-of-sample hold-out experiments, where the training data and the completely unseen test periods are strictly separated. In addition, it differs from the inner-loop cross-validation procedure employed in hyperparameter selection and fitting of DML nuisance models. The observations in the hold-out test set have never been used in the process of tuning, debiasing, and calculation of weights.

Due to the two-year lag restriction, this study has data available since 2017. Three successive unseen period hold-out tests are executed as follows (results in Table XV):

- Train window: 2017-2021 -> Unseen hold-out test period: 2022.
- Train window: 2017-2022 -> Unseen hold-out test period: 2023.
- Train window: 2017-2023 -> Unseen hold-out test period: 2024.

For each round, hyper-parameter selection, DML-based orthogonal de-biasing, and loss-weight computation (for both static stratified loss and time-varying fair loss) were performed using only samples from the corresponding training window. The hold-out test period remained completely unseen throughout the entire model-building procedure. Static stratified loss and the proposed time-varying fair loss were compared across these three fully unseen future test periods.

TABLE XV. UNSEEN-PERIOD HOLD-OUT TEST RESULTS: STATIC-STRATIFIED-LOSS VS TIME-VARYING FAIR LOSS

Hold-out Test Year	Metric	Static Stratified Loss	Time-Varying Fair Loss	Relative Reduction (Time-Varying vs Static)
2022	Overall RMSE	0.0283	0.0267	−5.65%
2022	Tier-3 RMSE	0.0346	0.0318	−8.09%
2023	Overall RMSE	0.0276	0.0261	−5.43%
2023	Tier-3 RMSE	0.0337	0.0309	−8.31%
2024	Overall RMSE	0.0279	0.0264	−5.38%
2024	Tier-3 RMSE	0.0341	0.0312	−8.50%

Note: Each hold-out test year represents a fully unseen future period. All weights, hyperparameters, and DML nuisance models are estimated only on training-window data without any access to hold-out observations. Tier-3 RMSE is reported because disadvantaged subgroups are most sensitive to temporal distribution shift.

As Table XV shows, in all three hold-out periods for the future that remain unseen (2022, 2023, and 2024), the time-varying fair loss results in lower RMSE than the static stratified loss. In terms of the unseen hold-out period, the total prediction error is decreased by about 5.4 to 5.7 %, whereas the most at-risk tier, Tier-3, experiences a decrease in error of 8.1 to 8.5 %. The results indicate that dynamic weight adjustment improves out-of-sample prediction performance across the three unseen periods.

H. Hyperparameter Sensitivity Analysis for Stratified Tier Weights

Sensitivity tests adjust the variance-derived tier weights by ±10% to examine model fairness stability. Fluctuations in Tier 3 aggregate RMSE remain below 2.5% after weight perturbation, confirming the data-driven loss framework is robust to minor weight adjustments and achieves a near-optimal accuracy-fairness tradeoff.

V. DISCUSSION

In sum, the proposed methodology performs well in terms of prediction accuracy and cross-tier fairness. Relative to vanilla Standard XGBoost, the static stratified custom loss alone achieves a 16.02% relative reduction in Tier-3 RMSE (evaluated on rolling-origin temporal-validation folds; Table IX, Fig. 5). Adding DML de-biasing yields a small incremental reduction in overall RMSE (−1.09 %) on top of the stratified-loss baseline. Subsequent introduction of the time-varying fair loss yields an additional overall RMSE reduction (−2.94 %) and a further small reduction in Tier-3 RMSE (−1.27 %). The DML orthogonal de-confounding yields conditionally de-biased associative estimates (TQ: 95 % CI $[0.132, 0.296]$; PCF: $[0.078, 0.216]$) under adjustment for observed confounders. Overfitting is avoided through Bayesian hyperparameter tuning, while time-varying loss makes RMSE less volatile in 2018 and 2022 shocks (Fig. 6 (a)), supporting fairness stability across the analytical period. The rankings identified by SHAP analysis (Fig. 8) include enrollment, infrastructure, moral education, teaching quality, and per capita funding as top predictors, indicating the joint contribution of institutional inputs and educational processes. Cross-fold SHAP CV is equal to 0.082 (CV < 0.1), which is 46.3% lower than in XGBoost, ensuring reliable policy implications.

The replicated nonlinear thresholds and interactions from Fig. 10 reveal tier-differentiated governance insights. From the LOWESS-smoothed SHAP dependence curve, this study empirically observes a predictive saturation point occurring at 0.85 for the teaching-quality score. This saturation point constitutes a descriptive empirical observation derived from SHAP partial-dependence visualization and is not estimated via formal regression-discontinuity-design inference. Beyond this predictive saturation point, marginal predictive gains for graduate employment outcomes diminish markedly. Based on the above empirical results and tiered fairness quantification shown in Table VI, the following targeted policy recommendations are suggested according to the three tiers of universities:

- Tier 3 vocational and applied universities: It is necessary to focus on increasing per capita funding and building campus infrastructure. Replicated error decomposition of the model shows that inadequate resource endowment is the main restriction for employment growth at lower-tier universities.
- Tier 2 provincial comprehensive universities: Teaching quality and external moral education require sustained investment to yield maximum marginal employment effect; these two process indicators demonstrate the largest positive SHAP contributions among mid-tier samples.
- Tier 1 Double First-Class universities: Teaching quality improvement provides decreasing marginal gains beyond the TQ saturation threshold. Industry-university cooperation and joint off-campus training can bring the largest increase in employment outcomes.

The aforementioned methodology can be applied to any hierarchical higher education system around the world. All core elements of this approach, namely DML debiasing, stratified dual-objective loss function, Bayesian hyperparameter tuning, time-varying fairness constraints, and IPO-SHAP analytics, can be fully replicated based on the balanced panel dataset.

The tier-weighted loss can be recalibrated according to within-group variance and generalized to different grouped prediction problems. The combination of DML correction and temporal fairness provides a standardized workflow capable of handling endogeneity and subgroup bias issues, thus giving a technical basis for empirical research on educational policy.

Furthermore, rolling-origin unseen-period hold-out tests (Table XI) verify that time-varying fair loss maintains superior predictive performance on completely unseen future time windows. Compared with static stratified loss, it mitigates performance decay induced by temporal distribution shift and delivers better out-of-sample generalisation for real-world forecasting scenarios.

VI. Conclusion

The objective of this study is to provide solutions for the challenges of graduate-employment prediction for multi-tier Chinese universities, involving observational-panel bias, cross-university predictive unfairness, and distribution drift in the year dimension. Within the Input-Process-Output theoretical framework, this research develops a time-varying DML-augmented stratified panel XGBoost system with time-varying fair loss, Bayesian hyperparameter tuning, and cross-fold SHAP interpretability components for prediction and policy-oriented analysis of university graduate employment performance.

In terms of overall prediction accuracy, the Enhanced XGBoost significantly outperforms the conventional linear models, Random Forest, GBM, and standard XGBoost on rolling-origin temporal-validation folds. Compared with standard XGBoost, the framework in this study reduces overall RMSE and produces a higher R^2 score, indicating improved prediction performance on balanced panel data of universities. The superior performance is further verified by the fully unseen future hold-out test set in 2022-2024.

For subgroup predictive fairness, the stratified and time-adaptive loss function substantially alleviates prediction bias across Tier-1, Tier-2, and disadvantaged Tier-3 universities. The static stratified loss provides a large relative error reduction for Tier-3 university subgroups. Furthermore, the time-varying fair loss dynamically adjusts subgroup penalty weights annually based on the residual variance in the training set and thus responds to temporal distribution changes. It ensures stable prediction performance of vulnerable university subgroups during macroeconomic shocks and consistently outperforms on unseen future test periods.

Thanks to the Double-Machine-Learning orthogonal debiasing approach, this research provides conditionally de-confounded associative estimations of teaching quality and per-capita funding with explicit identification assumptions based only on observable covariates and fixed effects. Note that these outputs are associative coefficients rather than causal treatment effects. Unobserved time-varying confounding cannot be eliminated by our workflow, which limits the extent to which causal interpretations can be drawn from the observational data.

With the help of cross-fold SHAP interpretability analysis, this study discovers the key predictors of graduate employment, among which enrollment, infrastructure capability, morality education, teaching quality, and per-capita funding have the strongest predictive contribution. Based on the LOWESS-smoothed SHAP dependence curves, the empirical predictive saturation thresholds of core process variables are found, which show the diminishing marginal predictive power of each variable after the turning points. The nonlinear and interaction patterns provide evidence for tier-specific higher education governance recommendations for three kinds of universities.

This study bears several important limitations:

- First, this work relies on a relatively small institutional-level sample, consisting of only 30 universities distributed across three tiers. Even though it covers different university types, the sample limits external generalizability; expanding multi-regional university datasets would strengthen out-of-sample validity. Moreover, applying the two-year lag for core predictors reduces the effective analytical sample from 300 raw observations (2015-2024) down to 240 usable records.
- Second, this is fundamentally an observational study design. While DML orthogonal de-biasing mitigates bias originating from observed confounders and fixed effects, unobserved time-varying confounding remains unaddressed. Full causal identification assumptions cannot be empirically verified within the present setup. In addition, the available time series yields a relatively short temporal horizon, which further constrains confident causal interpretation.
- Third, the dataset in this study is aggregated entirely at the university-institution level, with limited access to individual-level graduate characteristics. Consequently, individual-student heterogeneity that shapes graduate-employment outcomes cannot be explicitly incorporated into the model.

Several promising directions deserve future studies. First, the time-varying stratified fair-loss methodology can be generalized to various grouped panel prediction problems beyond the graduate employment forecast. Second, future studies can combine the student-level micro-data and the current DML-XGBoost pipeline to enable richer policy simulation. Third, algorithmic improvement can seek more balance between the global prediction accuracy and multiple group fairness. Additional repeated-seed simulation and leave-one-university-out cross-validation can further strengthen the robustness test.

VII. Declarations

A. Data Availability

The dataset employed in this study was collected from official statistical yearbooks and institutional annual reports

spanning 2015 to 2024, covering 30 Chinese universities categorized into three hierarchical tiers.

Dataset URL: https://doi.org/10.17632/shfxmjpycz.2

B. Author Contributions

Wang Yanlai: Conceptualization, methodology, software, formal analysis, writing—original draft, visualization.

Ting Tin Tin: Supervision, project administration, validation, writing—review and editing, funding acquisition, corresponding author.

Jia Jinlong: Data curation, investigation, resources, formal analysis.

Zhang Jingren: Data validation, formal analysis, investigation.

Xu Lin: Data validation, formal analysis, investigation.

ACKNOWLEDGMENT

1) Ethics statement: This study uses publicly available institutional panel data without involving human subjects. All data collection, sorting, and analysis comply with academic ethics and relevant regulations.

2) Competing interests: The authors declare no competing interests.

3) Funding statement: APC is supported by INTI International University and UNITAR International University.

REFERENCES

[1] Ahrens, A., Hansen, C. B., Schaffer, M. E., & Wiemann, T. (2025). Model averaging and double machine learning. Journal of Applied Econometrics, 40(3), 249-269. https://doi.org/10.1002/jae.3103

[2] Bai, Y., Wu, X., & Yang, A. (2024). Improved XGBoost-MLP model and application to performance prediction. In International Conference on Parallel and Distributed Computing: Applications and Technologies (pp. 414–426). Springer Nature Singapore. https://doi.org/10.1007/978-981-96-4207-6_38

[3] Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and machine learning: Limitations and opportunities. MIT Press.

[4] Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

[5] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, 21(1), C1–C68. https://doi.org/10.1111/ectj.12097

[6] Clarke, P. S., & Polselli, A. (2026). Double machine learning for static panel models with fixed effects. The Econometrics Journal, 29(1), 69–86. https://doi.org/10.1093/ectj/utaf011

[7] Feyzollahi, M., & Rafizadeh, N. (2026). Double/debiased machine learning for economists: Practical guidelines, best practices, and common pitfalls. Studies in Nonlinear Dynamics & Econometrics (SNDE). https://doi.org/10.1515/snde-2025-0011

[8] Ghazvini, A., Sharef, N. M., & Sidi, F. B. (2024). Prediction of course grades in a computer science higher education program via a combination of loss functions in an LSTM model. IEEE Access, 12, 30220–30241. https://doi.org/10.1109/ACCESS.2024.3376281

[9] Guevara-Reyes, R., Ortiz-Garcés, I., Andrade, R., Cox-Riquetti, F., & Villegas-Ch, W. (2025). Machine learning models for academic performance prediction: Interpretability and application in educational decision-making. Frontiers in Education, 10, 1632315. https://doi.org/10.3389/feduh.2025.1632315

[10] Ilgen, D. R., Hollenbeck, J. R., Johnson, M., & Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. Annual Review of Psychology, 56(1), 517–543. https://doi.org/10.1146/annurev.psych.56.091103.070112

[11] Jiang, K., Cai, X., Lv, Y., Xu, Y., Chen, Y., Wu, J., & Zheng, J. (2025). Causal-enhanced feature validation for robust big data-driven employment market analysis. INNO-PRESS: Journal of Emerging Applied AI, 1(2). https://doi.org/10.65563/jeaai.v1i2.14

[12] Jiang, Z., Shao, Z., & Wang, Q. (2025). Research on the employment mobility patterns and driving factors of graduates from double first-class universities in the central China. Scientific Reports, 15(1), 17373. https://doi.org/10.1038/s41598-025-71737-9

[13] Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper/2017/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html

[14] Lee, D. H., & Kim, T. H. (2020). Study on the prediction model for employment of university graduates using machine learning classification. The Journal of Information Systems, 29(2), 287–306. https://doi.org/10.5859/KAIS.2020.29.2.287

[15] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[16] Meng, L., Zhang, Y., Zhang, H., Li, J., & Liu, H. (2026). Analysis and prediction of university graduate employment based on machine learning. In Proceedings of the 2026 3rd International Conference on Informatics Education and Computer Technology Applications (pp. 70–74). https://doi.org/10.1145/3802133.3802146

[17] Merrick, L., & Taly, A. (2020). The explanation game: Explaining machine learning models using Shapley values. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction (pp. 17–38). Springer International Publishing. https://doi.org/10.1007/978-3-030-57321-8_2

[18] Neethling, L. (2023). Beyond enrolment: Academic incentives, outcomes and performance in higher education [Doctoral dissertation]. University of Cape Town. http://hdl.handle.net/11427/38524

[19] Nilforoshan, H., Gaebler, J. D., Shroff, R., & Goel, S. (2022). Causal conceptions of fairness and their consequences. In International Conference on Machine Learning (pp. 16848–16887). PMLR. https://proceedings.mlr.press/v162/nilforoshan22a

[20] Pelima, L. R., Sukmana, Y., & Rosmansyah, Y. (2024). Predicting university student graduation using academic performance and machine learning: A systematic literature review. IEEE Access, 12, 23451–23465. https://doi.org/10.1109/ACCESS.2024.3366789

[21] Pessach, D., & Shmueli, E. (2022). A review on fairness in machine learning. ACM Computing Surveys, 55(3), 1–44. https://doi.org/10.1145/3494672

[22] Ritonga, M., Rao, P. K., Alyamani, K. A. Z., Naved, M., Jawarneh, M., & Kumar, P. R. (2025). Analysis of graduate students using machine learning to forecast student's employability and academic success. In S. R. Sree & S. Kumar (Eds.), Algorithms and computational theory for engineering applications: ICACTEA 2024 (pp. 153–156). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-72747-4_23

[23] Saidani, O., Menzli, L. J., Ksibi, A., Alturki, N., & Alluhaidan, A. S. (2022). Predicting student employability through the internship context using gradient boosting models. IEEE Access, 10, 46472–46489. https://doi.org/10.1109/ACCESS.2022.3167666

[24] Scheerens, J. (2016). School effectiveness research and the development of a school effectiveness framework. Educational Research Review, 17, 1–10. https://doi.org/10.1016/j.edurev.2016.01.001

[25] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1), 148–175. https://doi.org/10.1109/JPROC.2015.2494226

[26] Su, C., Yu, G., Wang, J., Yan, Z., & Cui, L. (2022). A review of causality-based fairness machine learning. Intelligence & Robotics, 2(3), 244–274. https://doi.org/10.20517/ir.2022.17

[27] Sundararajan, M., & Najmi, A. (2020). The many Shapley values for model explanation. In International Conference on Machine Learning (pp. 9269–9278). PMLR. http://proceedings.mlr.press/v119/sundararajan20b.html

[28] Wang, Y., & Ting, T. T. (2026), "Public Information Data of 30 Representative Universities in China (2015-2024)", Mendeley Data, V2, https://doi.org/10.17632/shfxmjpycz.2

[29] Wen, M., Bastani, O., & Topcu, U. (2021). Algorithms for fairness in sequential decision making. In International Conference on Artificial Intelligence and Statistics (pp. 1144–1152). PMLR. https://proceedings.mlr.press/v130/wen21a.html

[30] Zhang, F., Xing, W., Li, C., & Jiang, Y. (2026). Fair AI in educational predictions: A multi-group fairness approach using reinforcement learning. The Internet and Higher Education, 101074. https://doi.org/10.1016/j.iheduc.2026.101074

APPENDIX

TABLE XVI. ABBREVIATIONS

Year	Tier	Residual Variance $\hat{\sigma}^2_{g,\text{train}}$	Raw Weight $\tilde{\omega}_g = 1/(\hat{\sigma}^2_{g,\text{train}} + \epsilon)$	Normalized Weight ω_g
2017	T1	0.00037	262.47	1
2017	T2	0.00059	167.52	1.57
2017	T3	0.00084	118.73	2.21
2018	T1	0.00035	276.12	1
2018	T2	0.00064	154.3	1.79
2018	T3	0.00091	109.77	2.52
2019	T1	0.00039	248.63	1
2019	T2	0.00061	162.24	1.53
2019	T3	0.00087	114.42	2.17
2020	T1	0.00041	236.41	1
2020	T2	0.00068	145.78	1.62
2020	T3	0.00096	104.25	2.27
2021	T1	0.00038	254.77	1
2021	T2	0.00063	157.46	1.62
2021	T3	0.00089	111.84	2.3
2022	T1	0.00042	230.87	1
2022	T2	0.00071	139.76	1.65
2022	T3	0.00099	101.31	2.28
2023	T1	0.00036	268.74	1
2023	T2	0.0006	164.81	1.63
2023	T3	0.00086	115.77	2.29
2024	T1	0.0004	242.78	1
2024	T2	0.00065	151.84	1.6
2024	T3	0.00092	108.49	2.24

Note: $\epsilon = 10^{-6}$ for numerical stability; variance and raw weights are estimated only from training-set data in each rolling-origin split to avoid information leakage. Normalized weight ω_g uses Tier 1 as baseline $\omega_{\text{Tier1}} = 1.00$. Larger normalized weight indicates stronger error penalty assigned to that subgroup in the stratified loss.

Design and Validation of an Agentless Web Application for IT Infrastructure Operations Management

Carlos Otárola-Mermau*, Alex Pacheco-Pumaleque
School of Systems Engineering, Universidad César Vallejo, Trujillo, Perú

***Abstract*—The manual management of information technology (IT) infrastructure operations at an IT services company in Lima, Peru, was characterized by fragmented procedures, delays, and limited traceability, which hindered service continuity and compromised information security. This study presents the design, development, and validation of an on-premises web application for the centralized management of IT infrastructure operations, evaluated across three dimensions: operational efficiency, execution effectiveness, and security with auditing. Built on a layered architecture, the solution employs Ansible as an automation engine that executes specific tasks on Linux, Windows, and VMware servers for diverse operations, without installing agents on the managed hosts. Access control is performed against Active Directory using roles that separate operations from administration. A pre-test/post-test quasi-experimental design compared the consolidated version of the application with its pilot version in a controlled test environment replicating the corporate infrastructure. The time reduction index increased by 41 percentage points, the operation success rate rose by 24 points, and the traceability level improved by 39 points, with Wilcoxon signed-rank tests confirming significance and large Cohen's effect sizes. The results confirm the viability of the solution and provide replicable evidence for the modernization of enterprise IT operations.**

Keywords*—*Web application; infrastructure orchestration; agentless automation; digital transformation

I. Introduction

The growing adoption of web solutions by organizations has become a key factor in the modernization of their services. Over the past decade, web applications have proven to be platforms capable of centralizing data and executing complex processes remotely and securely [1]. The interaction between software and technological infrastructure is now recognized as an essential component for ensuring business continuity; accordingly, the adoption of specific digital platforms has transformed operations management, moving from purely reactive approaches to strategic and proactive models [2].

Globally, information technology-related outages have been linked to human error and to ignored or inadequate procedures; nearly 40 percent of organizations have experienced outages caused by human error in the past three years [3]. Furthermore, the lack of centralized management tools increases the vulnerability of critical systems to operational failures [4]. In this scenario, digital transformation faces obstacles arising from the fragmentation of IT infrastructure and the slow adoption of integrated platforms, leading to recurring difficulties related to the lack of interoperability and the overloading of technical support channels.

From a theoretical perspective, process systematization is a key element for understanding how organizations succeed in transforming manual operations into structured, reproducible, and measurable workflows [5]. Along the same lines, sociotechnical systems theory holds that successful technology implementation requires simultaneous consideration of the technical and social subsystems, since their interaction determines the sustainability of organizational change [6]. Innovation theory, for its part, emphasizes that cloud computing, artificial intelligence, integrated platforms, and web applications act as transformation vectors with a significant impact on operational efficiency [7].

However, although technological solutions designed for IT service management exist, many lack the flexibility required to adapt to the administrative and regulatory workflows typical of corporate environments [8]. At a technology services company in Lima, Peru, the infrastructure department faced limitations stemming from the shortage of specialized personnel, the absence of clear policies, and deficiencies in process automation, which resulted in delays in task execution and in insufficient or nonexistent auditing mechanisms. Based on the analysis of the background and the theoretical foundations reviewed, the hypothesis was formulated that the implementation of a web application for IT infrastructure operations management would positively influence the organization's operational performance. Specifically, it was proposed that the application would improve time reduction rates, increase the task execution success rate, and optimize the level of event traceability.

In this context, the scientific contribution of this study is summarized in three contributions. First, an agentless orchestration architecture is proposed that centralizes the administration of heterogeneous IT infrastructure operations (Windows, Linux, and VMware servers) without requiring the installation of agents on the managed nodes, thereby reducing the attack surface and the administrative burden compared with agent-based approaches. Second, a traceability model—with audit logging per user, task, and host—is integrated into a single platform together with LDAP/Active Directory corporate authentication and role-based access control, capabilities that individual automation tools do not natively incorporate. Third, although some related works report quantitative improvements

*Corresponding author.

(Table I), these are limited to descriptive comparisons of indicators; this study additionally provides a quasi-experimental validation with inferential testing (Wilcoxon signed-rank test) and large effect sizes across the three indicators (r between 0.785 and 0.873, above Cohen's 0.5 threshold), which makes it possible to assess both the statistical significance and the practical relevance of the improvements in reducing response time, increasing the success rate, and enhancing traceability. Taken together, the contribution lies not in the individual tools—most of them open source—but in demonstrating that their orchestration under a coherent web architecture produces measurable and statistically significant operational improvements.

II. Related Work

Various studies show that process automation through web solutions significantly increases operational efficiency in business environments. For IT infrastructure monitoring, centralized platforms based on Prometheus and Grafana have been shown to provide real-time visibility of operational status, thereby reducing incident detection times and improving system availability [9]. Likewise, the adoption of service management frameworks such as FitSM, ITIL, COBIT, and ISO/IEC 20000 has been associated with measurable improvements in service quality and organizational operational efficiency [10]. Similarly, studies on intelligent automation platforms report incident resolutions up to 50% faster than traditional methods, attributable to the orchestration of repetitive tasks through automated workflows [11]. In the Peruvian mining industry, for example, web-based support systems have raised the incident resolution rate from 52% to 88% through ticket tracking and service level agreement monitoring [12]. Similarly, the digitalization of ITIL-based support services has improved management indicators, raising them from 55% to 82% following the adoption of digital processes [13].

Recent literature also highlights the advantages of agentless automation tools. In particular, research on Ansible demonstrates its ability to manage infrastructure through a declarative approach, which simplifies the management of IT configurations and operations, reducing costs and operational impacts [14], [15], [16]. However, these studies also show that, when operated independently from the command line, its use remains limited to specialized personnel, without integrated governance, traceability, or access control mechanisms for the rest of the organization.

Regarding access control, various studies have demonstrated the effectiveness of role-based models implemented with LDAP, as they enable scalable security and guarantee the functional separation between operational and administrative profiles [17], [18]. Table I presents a comparative analysis of the existing models and the proposed model.

Despite these advances, significant limitations persist. Consolidated orchestration platforms such as Ansible AWX/Tower [19] and Rundeck [20] offer centralized task execution but are centered on playbook management and scheduled jobs; they offer neither an integrated template catalog with categories and role-based separation of duties, nor a traceability model linking user, task, host, and execution output in a single auditable record, nor native hardware inventory modules with change detection. For their part, agent-based configuration management tools such as Puppet, Chef, and SaltStack [21] require deploying and maintaining persistent software on each managed node, which increases the operational burden and demands securing an additional component on every host. Moreover, many of these solutions are designed for cloud-native architectures, which conflicts with data sovereignty requirements in on-premises environments. In light of these gaps, this study proposes an on-premises web application that integrates, into a single platform, agentless orchestration governed by a template catalog, auditable traceability, and centralized corporate authentication.

TABLE I. Analysis of the Proposed Model and Related Work

Reference	Technology / Framework	Domain	Key benefits
Trikusuma et al., 2023 [9]	Grafana + Prometheus	IT Monitoring	Real-time visibility; reduction in Mean Time to Detection
Sarwar et al., 2023 [10]	FitSM, ITIL, COBIT, ISO/IEC 20000	Enterprise IT	Improves service quality and governance
Nugroho, 2022 [11]	Intelligent Automation Platforms	Enterprise IT	Results up to ~50% faster than manual methods
S. Perez and F. Mendoza, 2024 [12]	Web-Based Help Desk System	Mining company	Resolution rate: 52% → 88% through ticket tracking
Sánchez & Valles, 2021 [13]	ITIL v3 Implementation	Enterprise IT	Improved management indicators: 55% → 82%
Proposed Model, 2026	Web-based IT operations management platform (Flask, Ansible, LDAP, RBAC)	Enterprise IT	+41% reduction in time; +23.99% success rate; +39.17% traceability

III. Methodology

A. Research Type and Design

This is applied research, since its purpose is to solve a specific and verifiable problem: the fragmented, slow, and poorly traceable management of IT infrastructure operations at an IT services company, through the design and implementation of a software solution. The study was framed within the Design Science Research methodology, which establishes the construction of a technological artifact relevant to an organizational problem and its evaluation to demonstrate its usefulness [22]. The analysis is developed under a quantitative approach, using measurable and comparable indicators that allow an objective comparison between the initial condition and the one obtained after the intervention.

The design is quasi-experimental, with pre-test and post-test measurements applied to the same operational processes, without random assignment of units—as befits a business context where procedures and assets are predetermined [23]. The unit of analysis is IT infrastructure operations (for example, a password change on a group of servers, an NTP configuration, or security updates on a group of virtual machines); each operation was executed first with the pilot version of the application and then with the consolidated version. The sample

was census-based: the 30 operations comprising the infrastructure department's catalog of representative tasks were included, which eliminates selection bias. This within-subject comparison design reduces the variability attributable to differences between processes and isolates the effect of the technological intervention on the study's three indicators. The design is consistent with previous studies that evaluated web systems through before-after comparisons and reported significant improvements in resolution rates and service quality [12], [13]. To mitigate threats to internal validity, records were obtained directly from the system database, thereby avoiding manual transcription and observer bias.

The choice of a single-group design is grounded in operational feasibility criteria: in a real business environment, it is not possible to maintain two identical infrastructures operating in parallel, nor to subject the same assets to differentiated procedures simultaneously. Likewise, given the sample size and the non-normality of the data, the Wilcoxon signed-rank test was used, which is appropriate for small samples and non-normal distributions. Statistical analysis was performed with SPSS Statistics v25, applying the Shapiro-Wilk test to verify normality and the Wilcoxon signed-rank test with a significance level of 0.05; effect sizes were calculated using $r = Z/\sqrt{N}$ and interpreted according to Cohen's criteria.

B. Variables and Indicators

The independent variable is the web application, defined as the technological tool that centralizes and automates operations. The dependent variable is the management of IT infrastructure operations, operationalized in three dimensions, each with a quantitative indicator and a calculation formula, according to the study's consistency matrix. The operational efficiency dimension is measured by the response time reduction index (Ir), which expresses, as a percentage, the additional reduction in execution time achieved when moving from the pilot version of the application to its consolidated version. A higher value indicates a greater time reduction relative to the reference manual procedure. The execution effectiveness dimension is measured by the operation success rate (TasaE), which compares the number of completed operations with the total number of operations initiated, capturing the system's reliability in the face of partial failures or unreachable hosts. The security and auditing dimension is measured by the level of operational event traceability (NivelT), which quantifies the proportion of operations that generated a corresponding auditable record, reflecting the system's capacity to support subsequent control and accountability.

The three indicators are expressed on a ratio scale (0–100%) and are calculated using the formulas in Table II. Their selection is based on classifications of IT infrastructure performance indicators, which group the relevant metrics into availability, performance, security, and capacity, and on the concept of operational maturity as a key element for assessing the progress of digital transformation in the public and private sectors [24], [25]. It should be noted that the baseline measurements, recorded with the pilot version of the application, reflect the operating conditions prior to the intervention, which already included ad hoc scripts and partial command-line automation routines.

In this way, the model integrates the three dimensions into a single quantitative framework that can be replicated by other companies.

TABLE II. DIMENSIONS, INDICATORS, AND FORMULAS FOR THE DEPENDENT VARIABLE

Dimension	Indicator	Formula
Operational Efficiency	Time Reduction Index (Ir)	$Ir = ((Tmanual - Tauto) / Tmanual) \times 100$ [a]
Execution Effectiveness	Operation Success Rate (TasaE)	$TasaE = (CompletedOperations / TotalOperations) \times 100$
Security and Auditing	Traceability Level (NivelT)	$NivelT = (Log\_Entries / Operations) \times 100$

T_manual: time for the reference manual procedure; T_auto: time recorded by the application version corresponding to each phase (pilot in the pre-test, consolidated in the post-test).

C. Technology Selection

The selection of the technology stack was based on maturity and interoperability criteria, prioritizing components reliable for enterprise environments. The decision to implement a layered architecture was made on the principle of separation of concerns, which improved testability, facilitated the independent horizontal scalability of each layer, and reduced coupling between components [26]. Python was chosen as the main development language because it is a multi-paradigm language, with a moderate learning curve and an extensive standard library that facilitates integration with automation engines, network protocols, and database management systems [27]. Flask was adopted as the web framework for its lightweight core and modular extensions, which allow building scalable applications without imposing a rigid architecture [28]. Its support for the Model-View-Controller (MVC) pattern facilitates the separation between presentation, business logic, and data access [29], which is key for applications that require long-term maintainability. Flask also seamlessly integrates template rendering, database abstraction, and efficient HTTP request handling, accelerating development without sacrificing control over application behavior.

Ansible Core was chosen as the agentless automation engine for its declarative architecture, its ability to orchestrate tasks across multiple nodes simultaneously, and its native integration with Python through the ansible-runner library [30]. Unlike tools that require the installation of persistent agents on each managed node, it leverages protocols native to the operating systems themselves: Secure Shell (SSH), available by default on Linux distributions, and Windows Remote Management (WinRM), built into Microsoft Windows. By not requiring additional software on the managed nodes, this approach greatly simplifies initial deployment. The ecosystem was complemented with specialized libraries for data access, programmatic playbook execution, corporate authentication, vSphere integration, and credential protection, detailed in Table III.

Finally, PostgreSQL was selected as the relational database management system for its compliance with SQL standards, its native support for the JSONB data type—which allows storing semistructured data without sacrificing referential integrity—as well as its logical replication engine for high availability [31].

TABLE III. COMPLEMENTARY LIBRARIES OF THE TECHNOLOGY STACK

Component	Library	Purpose
Data access	SQLAlchemy 2 (ORM)	Database access layer
Automation engine	ansible-runner 2.4	Programmatic playbook execution
Corporate authentication	ldap3 2.9	Active Directory bind over LDAP
Virtualization	pyvmomi 8	VMware vSphere API integration
Credential encryption	cryptography 42 (Fernet AES-128-CBC)	Encryption of stored credentials
Password hashing	passlib (bcrypt)	Hashing of local passwords

D. Development Methodology

The web application was developed with an iterative and incremental approach, with progressive cycles adapted to the changing needs of the infrastructure. A hybrid methodology was adopted that combined the agile practices of Scrum—iterative sprints, a prioritized backlog, and retrospectives—with the risk management of the Rational Unified Process (RUP) [32].

The choice of this hybrid approach responded to three factors: the need for frequent testing cycles arising from the multiple integrations with heterogeneous systems (Active Directory, VMware vSphere, Linux, and Windows); the requirement for detailed technical documentation characteristic of an on-premises environment subject to audits, which purely agile methods tend to underestimate; and operation with limited resources, which made the full process overhead of RUP impracticable.

The backlog consisted of ten user stories estimated at 81 story points, distributed across six sprints: the first dedicated to the authentication core, and the remaining five to playbook execution, server management, the execution wizard, auditing, VMware integration, the playbook editor, internationalization, and exportable reports. Each increment was validated through integration testing against the laboratory environment servers, executing real operations through Ansible as the automation engine before acceptance.

The development cycle was organized into six phases; Fig. 1 shows the development flowchart.

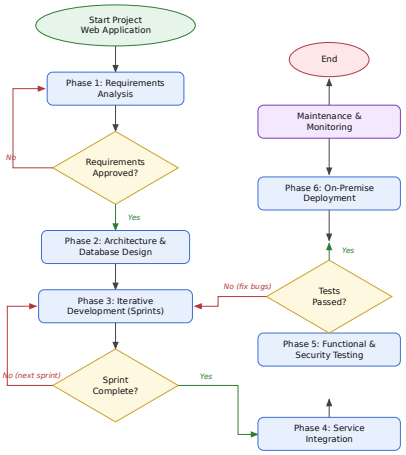


Fig. 1. Development flowchart.

1) Requirements analysis: Interviews Were Conducted with System Administrators and IT Operators to Identify Critical Manual Processes Amenable to Automation. The information was consolidated into a Catalog of Ten Functional Requirements and the Associated Non-functional Requirements—security, Performance, Auditability, and Usability—prioritized according to Their Operational Value and Technical Complexity, Which Constituted the Baseline Backlog for Development.

2) Architecture and database design: A Decoupled Layered Architecture—presentation, Application, Services, and Data—was defined, Which Allowed a Clear Distribution of Responsibilities and Improved System Maintainability; the Application Layer Was Organized into Independent Blueprints per Functional Domain (authentication, Operations, Administration, and REST API). the Database Model Was Designed Following Normalization Principles up to the Third Normal Form, Guaranteeing Referential Integrity and Minimizing Redundancy, with a Cross-cutting Auditing Component that Records Every Executed Operation.

3) Iterative development: The Project Was Structured into Six Two-week Sprints, according to the Thematic Distribution of the Backlog Described Above. Each Sprint Included Planning, Development, Unit Testing with Pytest, and Increment Review with Stakeholders, Whose Findings Fed Back into the Backlog of the Following Sprint.

4) Service integration: The Interoperability of the Application with the Environment's External Systems Was Consolidated: Centralized Authentication against Active Directory via LDAP, Infrastructure Inventory Management through the VMware vSphere API, and Programmatic Execution of Ansible Playbooks on the Managed Nodes. Each Integration Was Verified with Tests against the Laboratory Servers before Acceptance of the Corresponding Sprint.

5) Functional and security testing: The Acceptance Criteria of Each Module Were Verified, and a Security Testing Program Was Executed that Included: Brute-force Resistance on Authentication, with Account Lockout after Five Attempts; Privilege Escalation between Roles, Rejected with HTTP 403; Vulnerability Scanning with OWASP ZAP and SQL Injection Detection with SQLmap, with No Exploitable Findings; and Concurrent Load Testing with Locust. Additionally, Session Management (timeout, Secure Cookies) and At-rest Credential Encryption Were Validated.

6) On-premises deployment: The Application Was Deployed on Ubuntu Server 22.04 LTS as a Systemd Service, with Gunicorn 22 as the WSGI Server—with Workers Adjusted According to the Available vCPUs ($2 \times \text{vCPU} + 1$)—and Nginx as a Reverse Proxy for TLS Termination. Deployment Was Automated through Bash Scripts that Install Dependencies, Synchronize the Catalog Playbooks, and Restart the Service, with Database Backup and Log Rotation, Guaranteeing Reproducible Deployments.

E. System Architecture

The application was designed as an on-premises system structured in decoupled layers, deployed entirely within the company's internal network. The following description outlines the architecture from the user interface to persistent storage, reflecting the organization of the code in its current version; this architecture is illustrated in Fig. 2.

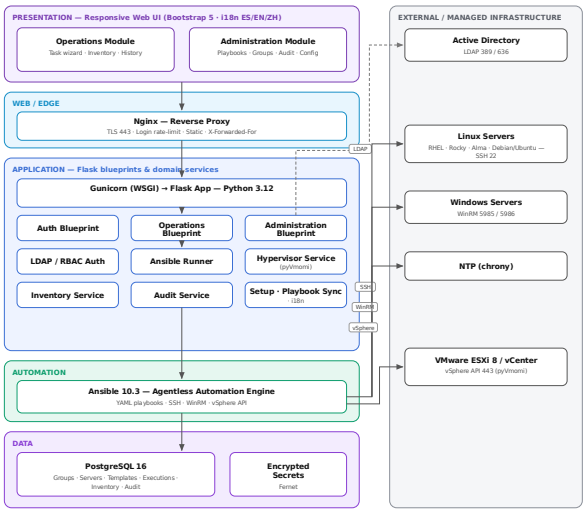


Fig. 2. System architecture.

1) Presentation layer: The Responsive Web Interface, Built with Bootstrap 5, Supports Multiple Languages (Spanish, English, and Chinese) and Strictly Separates the Operations and Administration Modules.

2) Web or edge layer: Nginx Acts as a Reverse Proxy and Single Entry Point. It handles TLS Encryption, Serves Static Files, Limits Login Attempts to Mitigate Brute-force Attacks, and forwards the Real Client Headers to Record IP Addresses for Auditing Purposes.

3) Application layer: Implemented in Python 3.12 with Flask and served by Gunicorn Behind Nginx. The logic is organized into Four Blueprints (authentication, Operations, Administration, and a REST API for External Integrations) and Several Specialized Services: LDAP/RBAC Authentication, Ansible Execution, VMware Hypervisor Management, Hardware Inventory, Event Auditing, Server Provisioning, Playbook Synchronization, and Internationalization. This Separation Facilitates System Extension and Maintenance.

4) Automation layer: Ansible, in Agentless Mode, Executes Declarative Playbooks on the Managed Hosts through the Native Protocols of Each Platform. The dynamic inventory is built from the Selected Servers and the Operator's Parameters, with Secure Credential Injection When Necessary.

5) Data layer: PostgreSQL Persistently Stores the System's Catalog, Executions, Inventories, and Audit Records; Sensitive Credentials Are Encrypted Using Fernet. External Integrations (Active Directory, Linux/Windows Servers, NTP, and VMware ESXi/vCenter) Remain Outside the Application and Are Accessible Only from the Internal Network, Preserving the On-premises Nature of the System.

F. Data Model

The relational model presented in Fig. 3 organizes the information around the entities that directly support the study's three indicators. Hypervisors constitute the root of the hierarchy: server groups (server_groups) and individual servers (servers) depend on them; the groups are composed of hosts homogeneous by platform and act as the target of operations.

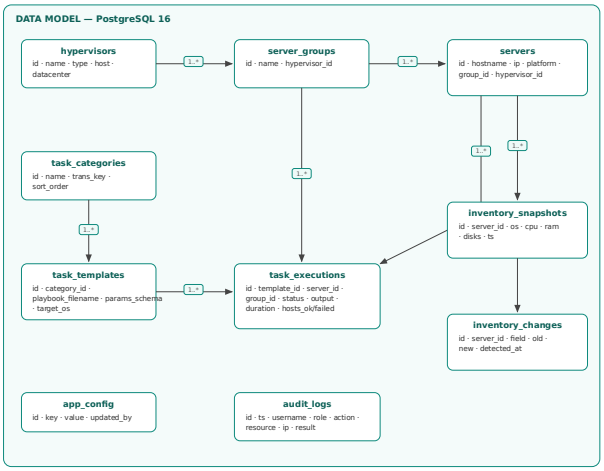


Fig. 3. Entity-relationship model of the PostgreSQL database.

For its part, the task catalog is structured into categories (task_categories) that contain templates (task_templates). Each template references a playbook, its parameter schema, and the target operating system, and must belong to a category to appear in the operations catalog. As for the transactional core, it consists of executions (task_executions): each record links the executed template, the target server or group, and the final status, storing the complete Ansible output, the duration, and the host metrics (ok, failed, unreachable) together with the parameters used. The indicators are naturally derived from this entity: time reduction index, success rate, and—together with the audit log (audit_logs)—the traceability level.

Finally, the inventory module is embodied in the snapshots (inventory_snapshots) and the detected changes (inventory_changes), both associated with the server, while the system configuration (app_config) consolidates the general parameters. This organization guarantees that every executed action leaves a structured, searchable trail—an essential requirement for subsequent auditing.

G. Authentication and Access Control

Access is authenticated against Active Directory using LDAP, avoiding credential duplication and leveraging the existing corporate identity. After authentication, the directory service retrieves the user's groups and applies role-based access control (RBAC): membership in the operators group enables the operations module, while membership in a specific administrators group (for example, group_admin_web) enables the administration module. Users without an authorized group are denied access, and the event is recorded as denied access (LOGIN_DENIED). As an additional safeguard against automated access attempts, the form incorporates the ALTCHA captcha—locally operated, based on proof of work—whose challenge is generated and verified with an HMAC key owned

by the application, without calls to external services, thus preserving the on-premises nature of the solution.

H. Operations Module and Task Catalogue

The operations module presents a three-step wizard—category, task template, and parameters/target group—that guides the operator without requiring command-line knowledge (Fig. 4). Form parameters are automatically transferred to Ansible, including the secure injection of hypervisor credentials when the operation involves VMware. The catalog is organized into five categories—password management, service configuration, virtualization, inventory, and maintenance—and includes, among others, validated tasks for password changes on Linux and Windows, time protocol configuration, on-demand restart of virtual machines on ESXi 8 and vCenter, hardware inventory collection, and operating system and SSL certificate updates. Operations requiring an irreversible step request explicit operator confirmation. The catalog is extensible: the administrator can incorporate new playbooks from the administration module, which become available to operators after validation. A read-only view of the hardware inventory was also incorporated into this module, allowing the operator to see the most recently collected status for each server without accessing the administration module.

I. Orchestration with Ansible and Traceability

When a task is confirmed, the system creates an execution record, logs the event in the audit log, and starts, on an asynchronous thread, the dynamic construction of the Ansible inventory from the hosts of the selected group; the client IP address is captured via ProxyFix and attached to the audited event. The engine is then invoked; upon completion, a PLAY RECAP parser extracts the host metrics (ok, failed, unreachable), determines the final status (success or failure), and persists the complete output, duration, and parameters. The interface allows viewing the complete Ansible output of each execution, which facilitates troubleshooting without having to access the server console. The inventory service analyzes the output of the collection playbooks and generates hardware snapshots—with change detection against the previous snapshot—that feed both the operator view and the administration dashboard.

J. Administration Module

The administration module—which is independent of the operations module and accessible only to the administrators group—organizes its functions in a collapsible accordion-style side menu; from there, YAML playbooks are edited with syntax validation, and—as a catalog governance requirement—both uploading a new playbook and editing an existing one require selecting one of the five available categories (password management, service configuration, virtualization, inventory, and maintenance), so that the resulting template appears classified under “Task Templates.” In addition, the module allows managing server groups and hosts by platform—including provisioning and automatic configuration of initial access via SSH/sudo—as well as managing templates and hypervisors and viewing inventory and audit reports; the latter incorporates a per-user filter and, like the inventory and execution history views, is displayed in paginated format. Finally, every relevant administrative action is recorded in the audit log, guaranteeing traceability and control over the operations performed.

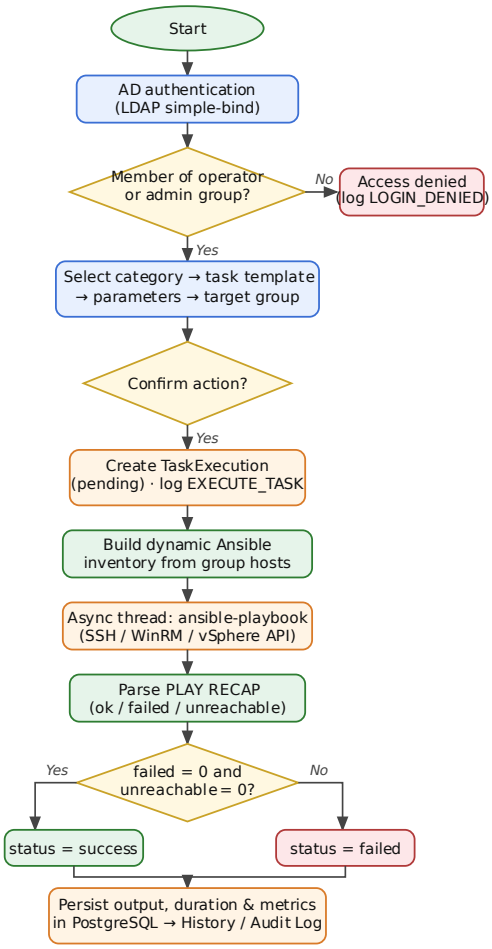


Fig. 4. Operator task execution flow.

K. Functional Validation Procedure

Since the project focuses on enterprise IT modernization, validation was performed in a controlled environment that replicated the target infrastructure using test virtual machines and development environments. The laboratory environment was deployed on VMware Workstation on a workstation with 24 CPU cores, 64 GB of RAM, and 2 TB of storage. The topology included ten virtual machines interconnected through an isolated internal virtual network: Linux servers (Oracle Linux, Ubuntu Server, and Debian), Windows Server machines, a VMware ESXi 8 host nested in standalone mode, and the virtual machine that hosted the web application. For each operation, two execution times were recorded: on the one hand, the execution time of the reference manual procedure—documented with IT personnel—and, on the other hand, the execution time automatically recorded by the web application in its version corresponding to each phase, stored directly in its database, which guaranteed consistent measurements free of transcription errors. In addition, the number of completed operations relative to the total initiated was counted, and the proportion of events in the audit log was verified. Data collection was supported by structured observation guides per indicator—one per phase, pre-test and post-test. The metrics

were calculated using the formulas in Table II, guaranteeing data traceability from the source to the indicator.

It should be noted that this was a laboratory environment and not the production infrastructure. This decision reflects security and operational ethics criteria—running experimental procedures on production systems would have put service continuity at risk—and is common practice in the evaluation of infrastructure automation systems. The environment faithfully replicated production conditions (same operating system distributions, same network topology, and same procedures documented with IT personnel), so the results are transferable to the real context, although this transferability is analyzed as a threat to external validity in the limitations section.

L. Security Considerations

Security was addressed comprehensively. Hypervisor credentials are stored encrypted using Fernet, and sensitive passwords are omitted from logs (no_log) to prevent them from leaking into log outputs. Communication with hosts uses SSH with a private key for Linux and WinRM for Windows, while the vSphere API is invoked over port 443. The number of concurrent executions is limited by a configurable semaphore that protects server resources. At the perimeter, Nginx handles TLS termination and limits the rate of login attempts; the form validates an ALTCHA captcha based on a proof-of-work mechanism—verified locally, without external services—and RBAC strictly separates operational and administrative privileges. Finally, for external connections, user access is routed through a VPN, guaranteeing that the application always operates within the internal network and is never directly exposed to the Internet.

IV. Results

A. Developed System

A fully functional on-premises web application was delivered that centralizes the management of IT infrastructure operations. The system integrates the two planned modules—operations and administration—as well as authentication against Active Directory with role-based access control, a catalog of automated tasks (Fig. 5), a hardware inventory module, and a comprehensive audit log. The administration dashboard consolidates in a single view the number of managed groups and servers, the available templates, the overall success rate, and the latest executions and audit events, eliminating the need to switch between disparate tools (Fig. 6).

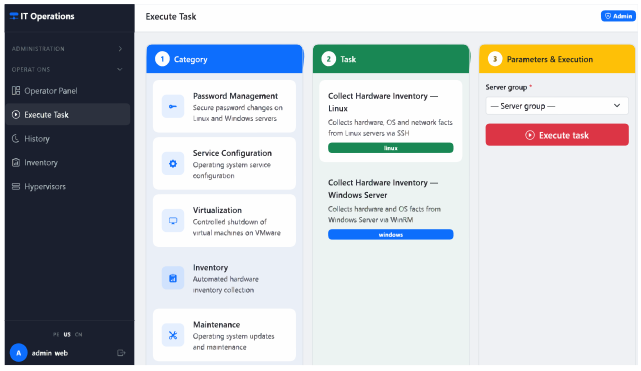


Fig. 5. Task execution wizard.

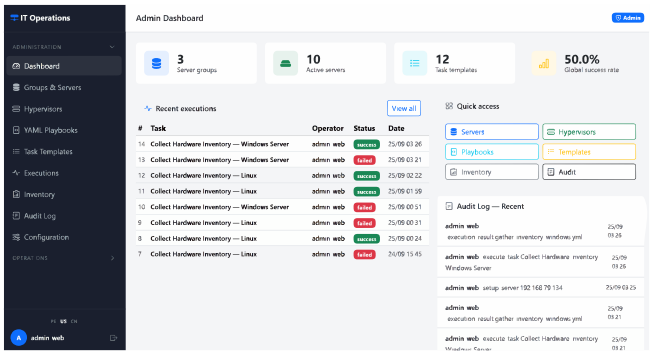


Fig. 6. Dashboard displaying consolidated operational metrics.

Group and server management organizes hosts by platform, shows their status, and allows additions, removals, and modifications, as well as automatic configuration of initial access. This organization directly feeds the dynamic construction of the Ansible inventory during each execution. Meanwhile, the hardware inventory module is now fully operational: once the collection playbook is executed, the system interprets its output and generates snapshots for each server, which are displayed both in the administration dashboard and in the operator's read-only view, thereby addressing a requirement that lacked systematic support in manual operations.

The audit log embodies the security and traceability dimensions: every significant action—logins, task executions, playbook edits, server provisioning, and denied access requests—is recorded with timestamp, user, role, action, affected resource, source IP address, and result, and can be filtered by user and paginated for review. This ability to reconstruct activity after the fact serves as direct input for the organization's internal control processes.

B. Operational Efficiency

TABLE IV. Absolute Execution Times by Phase and Condition

Phase	Condition	Mean (min)	SD	Min	Max
Pre-test	Reference manual procedure	53.90	6.76	45	65
Pre-test	Web application—pilot version	34.13	3.97	28	40
Post-test	Reference manual procedure	55.47	5.59	47	65
Post-test	Web application—consolidated version	12.97	3.35	8	18

Note: n = 30 operations per condition.

The descriptive analysis revealed substantial improvements across all dimensions. Table IV presents the absolute execution times recorded in each phase. In the pre-test, the pilot version of the application required an average of 34.13 minutes per operation (SD = 3.97), compared with 53.90 minutes for the reference manual procedure documented with IT personnel, which corresponds to the baseline time reduction index. In the post-test, the consolidated version reduced the mean time to 12.97 minutes per operation (SD = 3.35, maximum 18 minutes), while the times of the reference manual procedure remained stable between phases (53.90 versus 55.47 minutes), which rules out the improvement being due to a lower complexity of the

operations measured in the second phase. Moreover, the dispersion of times was notably reduced with the consolidated version (SD = 3.35), evidencing more predictable and homogeneous executions among operators.

From these measurements, the study's three indicators were calculated, whose descriptive statistics are presented in Table V. The time reduction index showed a positive difference of 41.00 percentage points, rising from a pre-test mean of 35.42% to a post-test mean of 76.42% with the consolidated version of the application. The operation success rate increased by 23.99 percentage points, from 70.74% to 94.73%. The event traceability level improved by 39.17 percentage points, from 55.02% to 94.19%.

TABLE V. DESCRIPTIVE STATISTICS FOR THE THREE INDICATORS

Indicator	Phase	N	Min	Max	Mean	SD
Ir (%)	Pre-test	30	11.11	56.25	35.42	12.65
Ir (%)	Post-test	30	66.04	86.15	76.42	6.47
TasaE (%)	Pre-test	30	62.86	80.77	70.74	5.05
TasaE (%)	Post-test	30	88.89	97.37	94.73	2.25
NivelT (%)	Pre-test	30	42.42	66.67	55.02	7.00
NivelT (%)	Post-test	30	89.66	97.50	94.19	2.60

The normality analysis using the Shapiro-Wilk test indicated that at least one measurement phase of each indicator did not meet the normality assumption ($p < 0.05$). Consequently, for methodological consistency, the nonparametric Wilcoxon signed-rank test for paired samples was applied in all three cases.

The statistical analysis confirmed significant improvements in the three indicators, with p-values below 0.001 and effect sizes classified as large according to Cohen's criteria. The magnitude of the standardized effect was greatest for the time reduction index, while the success rate and the traceability level showed comparable, albeit slightly smaller, effects. The detailed inferential statistics are summarized in Table VI.

TABLE VI. WILCOXON SIGNED-RANK TEST RESULTS

Indicator	Z	p-value	Effect Size (r)	Interpretation
Ir	-4.782	<0.001	0.873	Large
TasaE	-4.653	<0.001	0.850	Large
NivelT	-4.302	<0.001	0.785	Large

V. DISCUSSION

The results confirm that a web application that orchestrates tasks through Ansible significantly improves the three dimensions of enterprise IT infrastructure operations management. The improvement in time reduction of approximately 41 percentage points is consistent with the evidence that web platforms that automate technical tasks achieve notably faster resolutions than traditional methods [9], [11]. The improvement in the success rate, of approximately 24 percentage points, agrees with business studies reporting significant increases in incident resolution rates after the implementation of web-based systems [12], [13]. The improvement in traceability of approximately 39 percentage points directly addresses the security and auditing dimension that infrastructure performance indicator classifications identify as essential [24]. The large effect sizes (r between 0.785 and 0.873) of the three indicators demonstrate that the improvements are not only statistically significant but also relevant to business operations. It is worth noting that the success rate showed the smallest absolute gain (23.99 percentage points), which is partly explained by a ceiling effect: the pilot version already achieved a baseline performance of 70.74% and, in the post-test, values concentrated near 95%, where residual failures correspond to unreachable hosts or environment unavailability rather than to tool deficiencies; even so, its effect size was large ($r = 0.850$).

The hybrid development methodology that combines the iterative practices of Scrum with the documentary rigor of the Rational Unified Process proved effective for managing the complexity of enterprise infrastructure integrations [32]. The security validation program, which covered authentication, authorization, encryption, input validation, and session management testing, confirmed that the platform meets enterprise-level security requirements for on-premises deployment.

The immutable record of every action—including user, role, resource, and IP address—provides the audit trail that manual procedures could not guarantee, and offers the empirical evidence that studies on digital transformation in the public sector call for to assess progress toward modernization [10], [25].

Several limitations must be acknowledged. The single-group quasi-experimental design limited the ability to draw robust causal inferences, and the generalization of the findings to other corporate contexts is limited; both threats are examined in detail below according to the validity categories. The study was conducted in a development environment using virtual machines; deployment in production under sustained real business load is still pending. Additionally, because the baseline corresponded to the pilot version of the application itself, the measured effects represent the improvement attributable to the consolidation of the system and not the total impact relative to a purely manual process.

To systematize their analysis, the limitations are examined according to the four validity categories customary in empirical software engineering research. Regarding internal validity, the absence of a control group makes it impossible to completely rule out learning or history effects; to mitigate this threat, each operation was first measured with the pilot version as the baseline, and all measurements were automatically recorded in the system database, eliminating manual transcription and observer bias.

Regarding external validity, the results come from a single organization and a controlled development environment that replicated its operating conditions, which limits direct generalization. Nevertheless, the faithful reproduction of the environment—same operating system distributions, network topology, and procedures with IT personnel—favors the transferability of the findings to organizations with similar technological profiles.

Regarding construct validity, the three indicators (Ir, TasaE, and NivelT) operationalize IT operations management through verifiable formulas and ratio scales; however, they do not capture qualitative dimensions such as operator satisfaction or cognitive load, whose incorporation could be addressed in future work.

Finally, regarding conclusion validity, the use of the Wilcoxon test—appropriate for the sample size and the non-normality verified through Shapiro-Wilk—together with the reporting of effect sizes, reduces the risk of erroneous statistical conclusions.

VI. Conclusion and Future Work

This study designed, developed, and functionally validated an on-premises web application that improves the management of enterprise IT infrastructure operations. The application, built on a layered architecture using Ubuntu Server, PostgreSQL 16, Python 3 (Flask), and Ansible as the agentless automation engine, features authentication through the Lightweight Directory Access Protocol (LDAP) of Active Directory, role-based access control, and comprehensive audit logging.

Regarding the specific objectives, the solution increased the time reduction index by approximately 41 percentage points, raised the operation success rate by approximately 24 percentage points, and improved the event traceability level by approximately 39 percentage points. Wilcoxon signed-rank tests confirmed statistically significant improvements in the three indicators (p-values below 0.001, large effect sizes), with Z statistics ranging from -4.782 to -4.302. The detailed inferential results are presented in Table VI. The research went beyond the development of a software tool and provided empirical evidence that the systematic digitalization of enterprise infrastructure operations, adapted to the local corporate context, is a possible path to modernizing the technology industry.

The agentless architecture, validated by recent studies [15], [16], eliminates the overhead of agent management while maintaining scalability. The implementation of role-based access control, grounded in established access control models, ensures that administrative and operational functions remain properly segregated [33]. These findings provide a replicable framework for organizations seeking to modernize their information technology operations using agentless automation platforms.

A promising direction for future development is the implementation of a tiered change-approval workflow for significant infrastructure modifications. When an operator requests a high-impact change, the system could require a mandatory approval step by a user with an administrator profile before the automation playbook is sent to the execution engine. This governance layer aligns with recent frameworks that incorporate risk mitigation and internal controls into information technology operations, guaranteeing that critical changes receive adequate oversight and that accountability is maintained throughout the entire change life cycle [34].

Finally, as a broader extension, the incorporation of artificial intelligence-assisted validation of the Ansible playbooks uploaded to the platform is proposed. This component would automatically analyze files before execution to detect syntax errors, deviations from best practices, and potentially risky configurations, alerting the operator before the task reaches the execution engine. This line of work would connect the platform with recent AIOps trends, where language models and static analysis are combined to increase the reliability of infrastructure automation.

Another improvement involves strengthening the security layer through the integration of two-factor authentication based on time-based one-time passwords, which would further protect the platform against credential compromise. A systematic review of multifactor authentication methods confirms that time-based one-time password schemes remain one of the most practical and effective mechanisms for enterprise systems, and their inclusion complements the existing dual authentication model without disrupting the user experience [35].

References

[1] E. Veloz, V. Veloz, and D. Zamora, “Aplicaciones digitales como apoyo en la producción, agilidad y administración dentro de la empresa,” Revista Científica Ciencia y Tecnología, vol. 22, no. 36, Oct. 2022, doi: 10.47189/RCCT.V22I36.524.

[2] S. Nambisan, M. Wright, and M. Feldman, “The digital transformation of innovation and entrepreneurship: Progress, challenges and key themes,” Res. Policy, vol. 48, no. 8, p. 103773, Oct. 2019, doi: 10.1016/J.RESPOL.2019.03.018.

[3] Uptime Institute, “Uptime Institute’s 2022 Outage Analysis Finds Downtime Costs and Consequences Worsening as Industry Efforts to Curb Outage Frequency Fall Short.” [Online]. Available: https://uptimeinstitute.com/about-ui/press-releases/2022-outage-analysis-finds-downtime-costs-and-consequences-worsening

[4] D. Hajializadeh and M. Imani, “RV-DSS: Towards a resilience and vulnerability- informed decision support system framework for interdependent infrastructure systems,” Comput. Ind. Eng., vol. 156, p. 107276, Jun. 2021, doi: 10.1016/J.CIE.2021.107276.

[5] S. Angelopoulos, E. Bendoly, J. Fransoo, K. Hoberg, C. Ou, and A. Tenhiälä, “Digital transformation in operations management: Fundamental change through agency reversal,” Journal of Operations Management, vol. 69, no. 6, pp. 876–889, Sep. 2023, doi: 10.1002/joom.1271.

[6] M. Govers and P. Van Amelsvoort, “A theoretical essay on socio-technical systems design thinking in the era of digital transformation,” Gruppe. Interaktion. Organisation. Zeitschrift für Angewandte Organisationspsychologie (GIO), vol. 54, no. 1, pp. 27–40, Apr. 2023, doi: 10.1007/S11612-023-00675-8.

[7] A. A. Vărzaru and C. G. Bocean, “Digital Transformation and Innovation: The Influence of Digital Technologies on Turnover from Innovation Activities and Types of Innovation,” Systems, vol. 12, no. 9, 2024, doi: 10.3390/systems12090359.

[8] P. Kumar, N. Duraimurugan, and K. Ananthajothi, “Centralized Service Management Across Hybrid Environments: Harnessing Zookeeper with Cloud Computing,” 2024 International Conference on Smart Technologies for Sustainable Development Goals, ICSTSDG 2024, 2024, doi: 10.1109/ICSTSDG61998.2024.11026216.

[9] K. Trikusuma, V. Yasin, T. Budiman, A. Zulkarnain Sianipar, and A. Budi Yulianto, “IT Infrastructure Dashboard Monitoring Application Development Using Grafana and Promotheus, a Case Study at Astra Polytechnic School,” 2023 IEEE International Conference of Computer Science and Information Technology: The Role of Artificial Intelligence Technology in Human and Computer Interactions in the Industrial Era 5.0, ICOSNIKOM 2023, 2023, doi: 10.1109/ICOSNIKOM60230.2023.10364485.

[10] M. I. Sarwar, Q. Abbas, T. Alyas, A. Alzahrani, T. Alghamdi, and Y. Alsaawy, “Digital Transformation of Public Sector Governance With IT Service Management–A Pilot Study,” IEEE Access, vol. 11, pp. 6490–6512, 2023, doi: 10.1109/ACCESS.2023.3237550.

[11] H. Nugroho, Aradea, and K. Surendro, "IT Service Management Intelligence Model to Support the Implementation of Electronic Government System (EGS) in Indonesia," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 12, no. 2, pp. 494–500, Apr. 2022, doi: 10.18517/ijaseit.12.2.13789.

[12] S. Pérez y F. Mendoza, "Sistema Help Desk: una solución para la gestión de incidencias," Gaceta Científica, vol. 10, no. 2, pp. 76–81, mayo 2024, doi: 10.46794/gacien.10.2.2188.

[13] F. Sánchez and M. Valles, "Implementation of ITIL version 3 in organizations: Reasons for success and failure," Revista Científica de Sistemas e Informática, vol. 1, no. 2, pp. 54–67, Jul. 2021, doi: 10.51252/RCSI.V1I2.191.

[14] P. Patil, S. Phatkare, P. Rane, and S. Janathe, "Smart Infrastructure Management," Int. J. Innov. Sci. Res. Technol., vol. 11, no. 4, p. 18, May 2026, doi: 10.38124/IJISRT/26APR1306.

[15] R. Miñón et al., "ArtifactOps and ArtifactDL: a methodology and a language for conceptualizing and operationalising different types of pipelines," Journal of Cloud Computing, vol. 14, no. 42, Aug. 2025, doi: 10.1186/S13677-025-00761-W.

[16] R. Shrestha and A. A. N. Ali, "Configuration Management in Kubernetes Environments: A GitOps Approach," Proceedings - 2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing, UCC 2024, pp. 497–502, 2024, doi: 10.1109/UCC63386.2024.00077.

[17] A. K. Akuthota, "Role-Based Access Control (RBAC) in Modern Cloud Security Governance: An In-depth Analysis," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 11, no. 2, pp. 3297–3311, Apr. 2025, doi: 10.32628/CSEIT25112793.

[18] G. Fragkos, J. Johnson, and E. E. Tsiropoulou, "Centralized and Decentralized Distributed Energy Resource Access Control Implementation Considerations," Energies (Basel), vol. 15, no. 17, p. 6375, Sep. 2022, doi: 10.3390/EN15176375.

[19] Red Hat, "Ansible AWX Documentation," 2026. [Online]. Available: https://docs.ansible.com/projects/awx/

[20] PagerDuty, "Rundeck Documentation," 2026. [Online]. Available: https://docs.rundeck.com/

[21] J. O. Benson, J. J. Prevost, and P. Rad, "Survey of Automated Software Deployment for Computational and Engineering Research," in 2016 Annual IEEE Systems Conference (SysCon), Orlando, FL, USA: IEEE, Apr. 2016, pp. 1–6. doi: 10.1109/SYSCON.2016.7490666.

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research1," Management Information Systems Quarterly, vol. 28, no. 1, pp. 75–106, Mar. 2004, doi: 10.2307/25148625.

[23] D. I. K. Sjøberg et al., "Conducting realistic experiments in software engineering," ISESE 2002 - Proceedings, 2002 International Symposium on Empirical Software Engineering, pp. 17–26, 2002, doi: 10.1109/ISESE.2002.1166921.

[24] R. R. Fatkieva and A. S. Sudakov, "IT Infrastructure Performance Indicators," Scientific and Technical Information Processing, vol. 52, no. 1, pp. 38–44, Jun. 2025, doi: 10.3103/S0147688225700029.

[25] N. Fahmi, S. Murdha, and Suhardi, "Developing Digital Service Transformation Maturity Model in Public Sector," IEEE Access, vol. 12, pp. 174491–174506, 2024, doi: 10.1109/ACCESS.2024.3468341.

[26] Z. Tu, "Research on the Application of Layered Architecture in Computer Software Development," Journal of Computing and Electronic Information Management, vol. 11, no. 3, pp. 34–38, Nov. 2023, doi: 10.54097/JCEIM.V11I3.08.

[27] R. Dyer and J. Chauhan, "An exploratory study on the predominant programming paradigms in Python code," ESEC/FSE 2022 - Proceedings of the 30th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 684–695, Nov. 2022, doi: 10.1145/3540250.3549158.

[28] M. L. Hetland and F. Nelli, "Activity 3: Building a Web App with Flask," in Beginning Python: From Novice to Professional, M. L. Hetland and F. Nelli, Eds., Berkeley, CA: Apress, 2024, pp. 517–535. doi: 10.1007/979-8-8688-0196-9_27.

[29] M. Jannah, Dahlan, and C. Akbar, "Development of Management Information System Based on MVC Architecture to Improve Business Process Efficiency," Journix: Journal of Informatics and Computing, vol. 1, no. 2, pp. 63–76, Aug. 2025, doi: 10.63866/JOURNIX.V1I2.11.

[30] H. Sanugommula, "Exploration of the AWS Step Functions, Ansible, and AWS Lambda: Integrating Cloud Automation and Orchestration for Enhanced DevOps Efficiency," ESP Journal of Engineering & Technology Advancements, vol. 4, no. 3, pp. 108–110, 2024, doi: 10.56472/25832646/JETA-V4I3P109.

[31] M. Sheng, Y. Wang, C. Yang, S. V. Salunke, and A. Ouda, "A Performance Benchmark for the PostgreSQL and MySQL Databases," Future Internet, vol. 16, no. 10, p. 382, Oct. 2024, doi: 10.3390/FI16100382.

[32] J. Reiff and D. Schlegel, "Hybrid project management – a systematic literature review," International Journal of Information Systems and Project Management, vol. 10, no. 2, pp. 45–63, Jul. 2022, doi: 10.12821/IJISPM100203.

[33] J. S. Park and R. Sandhu, "RBAC on the Web by smart certificates," in Proceedings of the Fourth ACM Workshop on Role-Based Access Control, in RBAC '99. New York, NY, USA: Association for Computing Machinery, 1999, pp. 1–9. doi: 10.1145/319171.319172.

[34] O. H. Plant, J. van Hillegersberg, and A. Aldea, "Rethinking IT governance: Designing a framework for mitigating risk and fostering internal control in a DevOps environment," International Journal of Accounting Information Systems, vol. 45, p. 100560, Jun. 2022, doi: 10.1016/J.ACCINF.2022.100560.

[35] F. Sunny, A. Das, and J. Jobin, "A Review on Two-Factor Authentication Systems," International Journal of Engineering Research & Technology, vol. 11, no. 4, Jul. 2023, doi: 10.17577/IJERTCONV11IS04039.

The Impact of Gamified Escape Room Activities on Student Learning and Engagement in a Computer Database Course

Po Chan Chiu[1] *, King Kuok Kuok[2], Hamizan Sharbini[3],
Mohammad bin Hossin[4], Chih How Bong[5], Noor Hazlini Borhan[6]
Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak,
94300 Kota Samarahan, Malaysia[1, 3, 4, 5, 6]
Faculty of Engineering, Computing and Science, Swinburne University of Technology Sarawak Campus,
93350 Kuching, Malaysia[2]

***Abstract*—This study examines the impact of gamified escape room activities on student learning performance and engagement in a university-level database design course. A quasi-experimental repeated-measures design was used to compare the traditional learning approach with a gamified escape room incorporating quizzes, points, leaderboards, and nicknames. The gamified escape room featured defined goals, a game character, interactive tasks, immediate feedback, problem-solving challenges, and a fun learning environment. Results from the Wilcoxon Signed-Rank test showed significantly higher post-quiz scores and more consistent performance in gamified escape room-based quiz sessions. User engagement was assessed using the User Engagement Scale (UES), and students reported high levels of focused attention, perceived usability, aesthetic appeal, and reward. These dimensions align with the validated four-factor structure of the UES, which has shown good reliability and applicability across digital contexts. Aesthetic appeal received the highest rating, which implies that visually engaging and interactive design features played an important role in sustaining participation. Student feedback also indicated a strong preference for the gamified approach. These findings support the effectiveness of integrating gamification into classroom learning activities to enhance active learning, engagement, motivation, and knowledge retention.**

***Keywords*—Gamified learning; escape room; quiz; database course; engagement; motivation**

I. Introduction

The rapid advancement of digital technologies has transformed higher education, creating opportunities for innovative teaching approaches that support active and engaging learning experiences. Traditional lecture-based methods are often insufficient to sustain students' attention and participation. Consequently, educators are increasingly adopting student-centered pedagogical approaches that encourage interaction, collaboration, and experiential learning [1]. Among these approaches, gamification has gained prominence as a pedagogical approach to enhance student motivation, engagement, and improve learning outcomes in higher education [2, 3, 4].

Gamification refers to the application of game mechanics, such as points, badges, leaderboards, challenges, rewards and narratives to stimulate participation and improve user experiences [5]. Recent studies [6, 7, 8] have highlighted that gamification is an effective educational approach that encourages learners to solve problems and overcome challenges in game-based environments. Gamification offers innovative strategies to foster active learning and stimulate interest in academic content [9,10]. Inspired by this, escape room-based learning has been increasingly adopted in university settings to promote problem-solving, collaboration, knowledge retention and engagement through immersive and challenge-driven activities [11, 12].

Educational escape rooms transform traditional learning environments into interactive experiences where learners work under time constraints to solve content-related puzzles. These activities incorporate elements such as storylines, clues, and scoring, which have been shown to increase learners' engagement, motivation, and sense of accomplishment [13, 14]. When carefully aligned with learning objectives, such gamified methods can create deeper cognitive engagement, resulting in improved conceptual understanding and performance.

These advantages are particularly relevant to database education. Database courses require students to integrate theoretical knowledge with analytical reasoning and practical implementation while mastering complex concepts such as entity–relationship modelling, normalization, relational database design, SQL programming, and transaction management. Consequently, students frequently perceive database courses as abstract, cognitively demanding, and difficult to understand. While conventional paper-based quizzes remain a common method of content reinforcement, they often lack the dynamic interaction and immediacy that digital and game-enhanced activities can offer. To address these challenges, several studies have introduced gamification into database education. Schildgen [11] proposed MonsTER Park, an online educational game for learning Entity-Relationship Diagrams.

*Corresponding author.

Universiti Malaysia Sarawak and UNIMAS Scholarship of Teaching and Learning Research Grant (UNI/F08/SoTL-RG/85796/2023)

The study revealed that students enjoyed the game and did not encounter any difficulties in completing the tasks. However, the study did not explore the effectiveness of MonsTER in increasing students' learning engagement. Similarly, Dæhli et al. [12] developed a web-based tool known as LearnER to teach basic data modeling skills. The study demonstrated that the use of gamification elements such as earning points and a high score list in the LearnER tool significantly motivated the students and contributed to their learning of the data modeling course. The study reported that some students need more precise information on how to earn more points. Another important insight from the study is that more advanced game mechanics would further motivate the data modeling learning engagement process. In addition, the use of interactive digital learning environments has demonstrated positive educational impacts. For example, Deng et al. [13] found that Pencil Code, a web-based programming platform, significantly improved students' computational thinking skills and computer learning attitudes, highlighting the potential of technology-enhanced and interactive learning environments to foster deeper learner engagement and positive learning experiences. Furthermore, Barra et al. [14] investigated the use of a remote educational escape room for database learning across four higher education courses over two academic years. Using pre- and post-tests, student perception questionnaires, and learning analytics collected through a web platform, the study demonstrated that remote educational escape rooms were both engaging and effective for teaching database concepts. Students reported high levels of enjoyment, immersion, and satisfaction, with more than 95% recommending that similar activities be incorporated into other courses. The findings suggest that educational escape rooms provide an effective approach for reinforcing theoretical database concepts while simultaneously developing collaborative problem-solving skills.

Although gamification has been explored in various academic contexts, limited research has focused on the effects of escape room-based quizzes on learning performance and engagement in university-level computer database courses. Database knowledge is essential in IT education. The database course was one of five information technology pillars in the 2008 ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programs in Information Technology [15]. Furthermore, the most recent version of the ACM/IEEE Computer Society Task Group on Information Technology Curricula (2017) includes data modeling and database query languages as essential IT topics in information management [16].

Therefore, this study addresses the gap by evaluating how traditional and gamified escape room-based quiz learning activities influence students' post-lecture performance and their perceptions of learning within a database design course. Specifically, the study aims to examine the effectiveness of employing a gamified escape room on students' learning performance and engagement. To evaluate students' engagement levels, the User Engagement Scale (UES) was adopted as a measurement instrument to assess dimensions such as focused attention (FA), perceived usability (PU), aesthetic appeal (AE), and reward (RW). By incorporating both learning performance and engagement measures, this study provides a more comprehensive understanding of the effectiveness of gamified learning activities in database education.

To guide this investigation, the primary research questions are formulated:

- Is there any significant difference in students' learning performance between paper-based quizzes and gamified escape room-based quizzes?
- To what extent do gamified escape room activities promote student engagement in terms of FA, PU, AE, and RW?

The main contributions of this study are twofold. First, the study provides empirical evidence on the effectiveness of gamified escape room-based quizzes in improving students' learning performance compared with traditional paper-based quizzes. The Wilcoxon Signed-Rank test results demonstrate significant improvement in students' post-lecture performance, which implies the effectiveness of gamified escape room activities as an active learning approach. Second, the study evaluates student engagement with the gamified escape room using the UES across four dimensions: FA, PU, AE, and RW. The findings reveal high levels of engagement across all four dimensions, with aesthetic appeal receiving the highest rating. This demonstrates that the gamified escape room provides an engaging and positive learning experience for students.

II. Material and Methods

A. Participants and Setting

The participants in this study were 200 undergraduate students enrolled in a Database Design course at a Malaysian higher education institution. All participants were aged between 18 and 26 years old. Before the study began, all students were briefed on the study's objectives and procedures and provided informed consent. The learning sessions took place in a face-to-face classroom setting and incorporated two types of structured learning activities: traditional paper-based quizzes and gamified escape room-based quizzes (gamified ER-based quizzes). These quizzes were not graded in the course assessment and were used purely as active learning tools to reinforce students' understanding of the lecture content and support knowledge retention.

B. Learning Tools

Traditional Quizzes: These were paper-based quizzes consisting of multiple-choice questions (MCQs) designed to assess students' understanding of key concepts presented during the lectures.

Gamified Escape Room Quizzes: These quizzes were delivered through an interactive digital escape room platform. Students solved a series of MCQ quizzes and challenges based on the lesson content. Elements of gamification such as leaderboards, badges, points, scores, and nicknames were embedded to enhance the student learning experience and motivation.

C. Study Design

This study employed a quasi-experimental repeated-measures design conducted over a period of four consecutive weeks. The same group of students participated consistently throughout the intervention. Fig. 1 illustrates the four-week study design combining lectures, quizzes, and gamified activities. Each week consisted of a scheduled face-to-face lecture accompanied by a pre-lecture and post-lecture quiz.

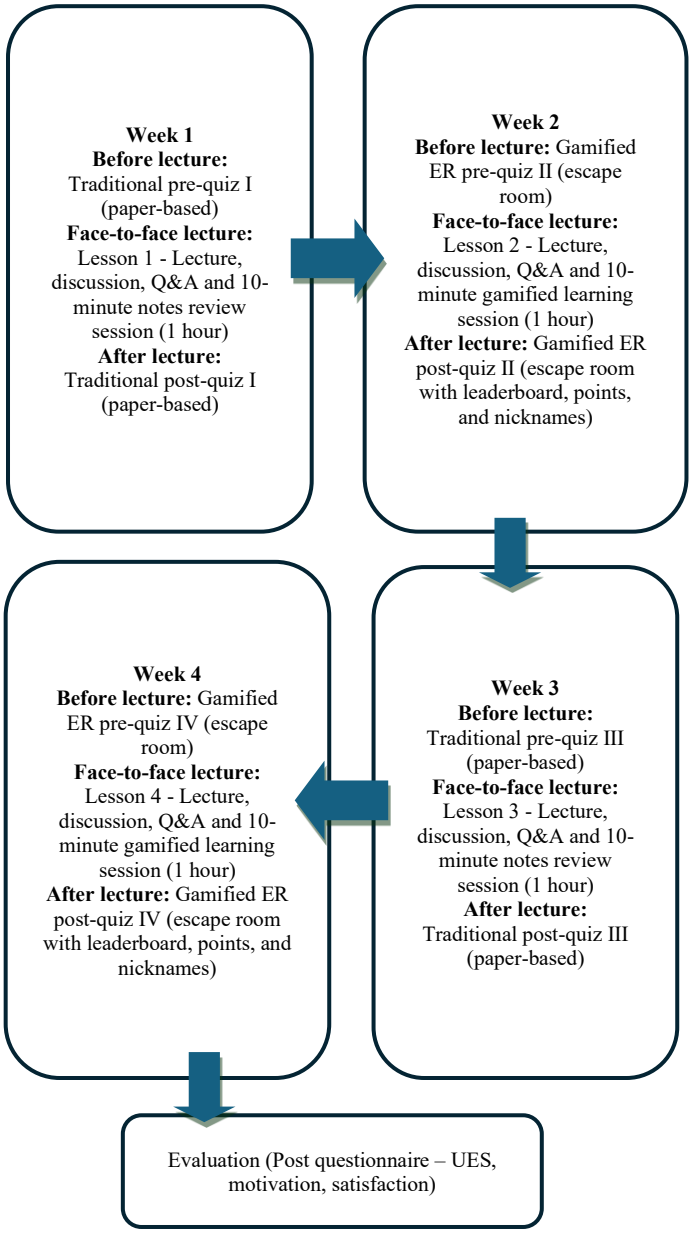


Fig. 1. Study design.

The pre- and post-quizzes alternated between traditional paper-based and gamified ER-based quizzes across the weeks to allow comparison. The quizzes were designed to have comparable levels of difficulty across both formats. Quantitative data were collected from the pre- and post-quiz scores to measure students' learning performance.

In addition, a post-intervention questionnaire was conducted after the Gamified ER post-quiz IV (Week 4) to evaluate student engagement and satisfaction across the different learning tools. The learner engagement data were also collected using the adapted User Engagement Scale (UES) developed by O'Brien et al. [17]. The UES assesses engagement across four constructs: Focused Attention (FA), Perceived Usability (PU), Aesthetic Appeal (AE), and Reward Factor (RW). The items were measured using a 5-point Likert scale ranging from strongly disagree to strongly agree. The collected evaluation data were analysed using descriptive statistics to summarise engagement levels across the UES dimensions. Furthermore, the reliability of the adapted UES was evaluated using Cronbach's alpha, following the approach adopted by Amriza et al. [18].

III. Gamified Platform

A. Gamification Elements

Building on previous research that identified common gamification elements in educational settings [15, 19, 20], this study integrated several features to boost student motivation and engagement. Specifically, the gamification elements included a leaderboard, escape room challenge, background music, points system, and digital certificates. These features were embedded within the gamification learning platform and deployed during the in-class gamified ER-based quiz sessions. The escape room activity served as the core interactive component, where students solved database subject-related challenges. Background music, which could be turned on or off by the students, provided an immersive and stimulating environment. Points were awarded based on task completion and accuracy, and students' cumulative scores were reflected in a dynamic leaderboard visible to all participants. This encouraged healthy competition among peers. To preserve anonymity and promote fairness, students participated under nicknames rather than real names. At the end of the session, students received digital certificates as recognition of their performance. Their ranking position and the top three scores remained visible on the gamification learning platform until the next scheduled activity to provide sustained motivation and a sense of accomplishment.

B. Gamified Learning Session

As illustrated in Fig. 2 and Fig. 3, the gamified learning session was designed as an engaging learning activity to reinforce key concepts covered in the database course. The session consisted of 10 interactive questions delivered through a game-based interface. The question types varied and included MCQs, true or false, and fill-in-the-blank formats. This variety aimed to support content recall and conceptual understanding through active participation. The interface featured a visually immersive environment with game-like elements such as a score tracker and a themed background. The session served as a preparatory activity before students advanced to the escape room quiz, which offered further opportunities to apply their knowledge in a more challenging and problem-solving context.



Fig. 2. Gamified learning platform.



Fig. 3. Gamified learning question.

C. Escape Room

The escape room approach in education is a modern pedagogical strategy designed to motivate students through immersive and interactive learning experiences. Prior studies have demonstrated its effectiveness in enhancing learning across subjects such as Mathematics [21,22]. Drawing inspiration from these, the gamified escape room in this study integrates goal-setting, game characters, interactive tasks, feedback, problem-solving, and a fun learning environment.

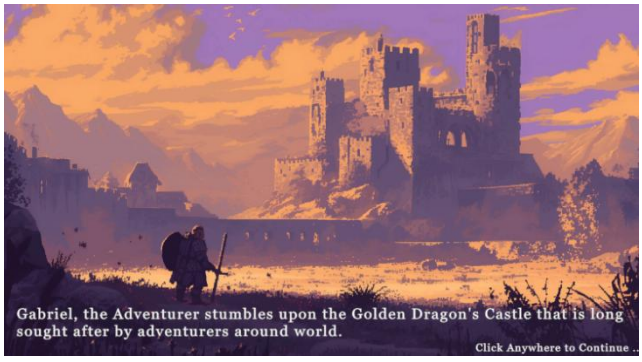


Fig. 4. Gabriel, the adventurer, enters the castle to seek treasure.

As illustrated in Fig. 4, the storyline begins with Gabriel, the adventurer, entering the Golden Dragon's Castle in search of treasure. Students solve database challenges to help Gabriel progress through the escape room adventure.



Fig. 5. Escape Room (ER) page.

Fig. 5 shows the escape room setting where learners engage with the environment to unlock clues and progress in the game. Each challenge is tied to database concepts, encouraging students to analyze clues, recall lesson content, and apply knowledge in a meaningful way, which mirrors the real-world problem-solving in database design (see Fig. 6). After completing each challenge, students receive a clue that helps them unravel the final escape code.



Fig. 6. Gamified ER-based quiz.

IV. Results

A. Student Learning Performance

The findings of this study reveal that the gamified escape room learning activity significantly enhanced students' learning performance compared to the traditional paper-based quiz activity. All students participated in both activities, enabling a within-subject comparison of their performance.

TABLE I. Wilcoxon Signed-Rank Test for Comparing Students' Achievements Between Paper-Based and Gamified Escape Room-Based Post-Quizzes

Quizzes	N	Wilcoxon Signed-Rank Test			
		Mean	*W-value*	*z-score*	*p-value*
Paper-based quiz I	200	80.58	172.50	10.85	< .001
Gamified ER-based quiz II	200	98.67			
Paper-based quiz III	200	85.17	0.00	9.42	< .001
Gamified ER-based quiz IV	200	99.83			

As shown in Table I, the Wilcoxon Signed-Rank test results indicated statistically significant differences in student scores between the two types of learning activities. Following the face-to-face lecture in Lesson 1, which included a lecture, discussion, question-and-answer session, and a short 10-minute notes review, students completed the Paper-based Post-Quiz I. The mean score obtained for this post-quiz was 80.58. In contrast, in Week 2, the face-to-face lecture for Lesson 2 incorporated a 10-minute gamified learning session, discussion, and question-and-answer activities, followed by gamified ER Post-Quiz II. The mean score increased to 98.67, which was significantly higher than the paper-based Post-Quiz I mean score of 80.58 ($W = 172.50$, $z = 10.85$, $p < .001$). A similar trend was observed in Weeks 3 and 4, where the paper-based Post-Quiz III (Lesson 3) produced a mean score of 85.17, whereas the gamified ER Post-Quiz IV achieved a significantly higher mean score of 99.83 ($W = 0$, $z = 9.42$, $p < .001$).

On average, students completed the escape room challenge in approximately six minutes. These results underscore the added instructional value of combining gamified learning with interactive activities like escape rooms, compared to traditional passive review.

TABLE II. COMPARISON OF STUDENTS' PRE- AND POST-QUIZ PERFORMANCE FOR PAPER-BASED AND GAMIFIED ESCAPE ROOM-BASED QUIZZES

Quizzes	N	Pre-Quiz			Post-Quiz		
		Mean	*SD*	*Mdn*	*Mean*	*SD*	*Mdn*
Paper-based quiz I	200	23.83	16.71	16.67	80.58	16.31	83.33
Gamified ER-based quiz II	200	27.58	16.21	33.33	98.67	4.52	83.33
Paper-based quiz III	200	25.25	15.9	16.67	85.17	14.79	100
Gamified ER-based quiz IV	200	34.42	14.15	33.33	99.83	1.66	100

Further evidence can be seen in the pre- and post-quiz descriptive statistics in Table II. Students began with similar pre-quiz medians (16.67 to 33.33), reflecting comparable baseline knowledge. After the learning sessions, post-quiz medians increased substantially, reaching 83.33 for both Quiz I and Quiz II, and 100 for Quiz III and Quiz IV. Additionally, the standard deviations were lower in the gamified ER-based quiz activities (e.g., SD = 4.52 for Quiz II and 1.66 for Quiz IV), indicating more consistent performance, possibly due to the motivational and interactive elements embedded in the gamified learning environment.

These quantitative findings were further reinforced through student perceptions. All respondents expressed high levels of satisfaction with escape room learning and recommended it as a supplementary tool for juniors in database courses. All students indicated a preference for the gamified ER-based quizzes approach over the conventional paper-based quizzes. They cited its greater satisfaction, motivation and effectiveness during class activities. This makes us believe that interactive and gamified learning is not just effective in terms of academic performance but is also well-accepted and pedagogically impactful.

B. Student Engagement and Experience

After the gamified ER post-quiz IV, all participants completed the UES O'Brien et al. [17] to evaluate their experience with the gamified ER-based quiz activities (see Appendix- Table VI). The questionnaire items use a 5-point Likert scale (1 "Strongly Disagree" to 5 "Strongly Agree"). The UES evaluates four engagement dimensions: Focused Attention (FA), Perceived Usability (PU), Aesthetic Appeal (AE), and Reward (RW).

To assess the UES evaluation results, the response values for the UES items PU.1, PU.2, PU.3, PU.4, PU.5, PU.6, PU.8, and RW.3 were reverse-coded before analysis [23]. Then, descriptive statistics were calculated to summarize participants' responses for each item, including the mean, standard deviation (SD), minimum, median, maximum, and mode. As shown in Table III, the mean scores across all constructs were high, ranging from 4.41 to 4.74. These high mean scores indicate a high level of student engagement with the Gamified ER-based quiz activities. Most items also recorded a mode of 5, in which the majority of participants strongly agreed with the statements related to engagement, usability, visual appeal, and perceived reward for the activity. These findings suggest that the gamified escape room design effectively supported active participation, high engagement, and positive learning experiences among students.

TABLE III. DESCRIPTIVE STATISTICS OF UES EVALUATION RESULTS (N = 200). ITEMS WERE RATED ON A 5-POINT LIKERT SCALE (1 = STRONGLY DISAGREE, 5 = STRONGLY AGREE)

UES	*Mean*	*SD*	*Min*	*Mdn*	*Max*	*Mode*	*N for Mode*
FA.1	4.69	0.46	4	5	5	5	138
FA.2	4.62	0.62	2	5	5	5	133
FA.3	4.61	0.62	2	5	5	5	132
FA.4	4.61	0.67	2	5	5	5	135
FA.5	4.56	0.74	2	5	5	5	132
FA.6	4.61	0.57	2	5	5	5	128
FA.7	4.6	0.63	2	5	5	5	129
PU.1	4.53	0.58	2	5	5	5	113
PU.2	4.41	0.67	2	4	5	5	97
PU.3	4.44	0.6	2	4	5	4	100
PU.4	4.46	0.6	2	5	5	5	100
PU.5	4.49	0.6	2	5	5	5	106
PU.6	4.49	0.57	2	5	5	5	103
PU.7	4.61	0.59	2	5	5	5	129
PU.8	4.46	0.6	2	4	5	5	99
AE.1	4.74	0.44	4	5	5	5	147
AE.2	4.71	0.45	4	5	5	5	142
AE.3	4.69	0.55	2	5	5	5	144
AE.4	4.68	0.56	2	5	5	5	143
AE.5	4.63	0.55	2	5	5	5	131
RW.1	4.46	0.68	2	5	5	5	105
RW.2	4.64	0.49	3	5	5	5	128

RW.3	4.45	0.61	2	4	5	5	99
RW.4	4.59	0.63	2	5	5	5	128
RW.5	4.66	0.57	2	5	5	5	139
RW.6	4.68	0.58	2	5	5	5	143
RW.7	4.64	0.54	2	5	5	5	131
RW.8	4.68	0.5	2	5	5	5	137
RW.9	4.63	0.54	2	5	5	5	130
RW.10	4.67	0.56	2	5	5	5	139

Note: SD = standard deviation; Min = minimum; Mdn = median; Max = maximum

C. Reliability Analysis and Descriptive Statistics

Reliability analysis and descriptive statistics were conducted to assess the reliability of each UES scale. As shown in Table IV, all constructs demonstrated good internal consistency, with Cronbach's alpha values ranging from 0.840 to 0.913. The mean scores for the UES scales were also relatively high, ranging from 4.484 to 4.689. Among the dimensions, AE recorded the highest mean score (Mean = 4.689), followed by RW (Mean = 4.607), FA (Mean = 4.612), and PU (Mean = 4.484). These results suggest that students perceived the gamified ER activity as visually appealing, rewarding, and capable of sustaining their attention. The relatively small standard deviations further indicate that the responses were consistent across participants. Overall, the findings show positive student engagement in the gamified ER activity. It also demonstrates that the UES scales are reliable measures for evaluating different dimensions of student engagement.

D. Validity Analysis

Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) were further conducted to examine the factor structure. As shown in Table V, the results supported a four-factor model consisting of FA, PU, AE, and RW. All factor loadings were high and statistically significant ($p < .001$), indicating that the UES questionnaire is a reliable instrument for measuring student engagement in the gamified ER environment.

TABLE IV. RELIABILITY ANALYSIS AND DESCRIPTIVE STATISTICS

UES	No Items	Cronbach's Alpha	Mean	Standard Deviation
Focused Attention (FA)	7	0.840	4.612	0.445
Perceived Usability (PU)	8	0.913	4.484	0.476
Aesthetic Appeal (AE)	5	0.903	4.689	0.438
Reward (RW)	10	0.881	4.607	0.399

TABLE V. FACTOR LOADINGS OF THE GAMIFIED ER WITH 1-FACTOR EXPLORATORY MODEL AND WITH 4-FACTOR CONFIRMATORY MODEL

Factor		Exploratory Factor Analysis (EFA)	4-factor Confirmatory Factor Analysis (CFA)					
			FA	*PU*	*AE*	*RW*	*z-score*	*p-value*
Focused Attention (FA)	FA.1	0.850	0.869				25.805	< .001
	FA.2	0.921	0.861				31.983	< .001
	FA.3	0.893	0.796				23.726	< .001
	FA.4	0.810	0.757				18.748	< .001
	FA.5	0.858	0.827				25.073	< .001
	FA.6	0.684	0.879				30.290	< .001
	FA.7	0.607	0.860				25.901	< .001
Perceived Usability (PU)	PU.1	0.849		0.830			28.275	< .001
	PU.2	0.893		0.871			40.602	< .001
	PU.3	0.953		0.953			63.849	< .001
	PU.4	0.920		0.975			56.664	< .001
	PU.5	0.921		0.925			49.931	< .001
	PU.6	0.883		0.926			53.988	< .001
	PU.7	0.467		0.567			8.716	< .001
	PU.8	0.891		0.855			34.431	< .001
Aesthetic Appeal (AE)	AE.1	0.874			0.951		63.424	< .001
	AE.2	0.872			0.937		59.547	< .001
	AE.3	0.961			0.951		70.568	< .001
	AE.4	0.913			0.952		47.846	< .001
	AE.5	0.936			0.957		64.474	< .001
Reward (RW)	RW.1	0.570				0.636	13.771	< .001
	RW.2	0.923				0.924	46.104	< .001

	RW.3	0.519				0.653	14.906	< .001
	RW.4	0.872				0.852	30.538	< .001
	RW.5	0.785				0.778	25.107	< .001
	RW.6	0.805				0.757	21.714	< .001
	RW.7	0.884				0.873	35.082	< .001
	RW.8	0.880				0.900	47.036	< .001
	RW.9	0.905				0.921	44.950	< .001
	RW.10	0.903				0.906	39.562	< .001

V. DISCUSSION

These findings are consistent with previous research highlighting the benefits of gamified instructional approaches in higher education settings. Zainuddin et al. [24] emphasized that integrating interactive and competitive elements through gamification can enhance learning outcomes. In the present study, the use of escape room–based quizzes enabled students to engage with database concepts in a simulated and challenge-driven environment that mirrored active learning principles. The gamified activities likely helped students score higher in both gamified sessions (Quiz II and Quiz IV). This pattern was also observed in a similar study by Gordillo and López-Fernández [25], which demonstrated increased knowledge acquisition and improved student perceptions in software engineering education through escape rooms.

The reduced standard deviations in the gamified ER-based quiz results (SD = 4.52 and 1.66) suggest that students performed more consistently compared to the traditional paper-based quizzes. This indicates that the escape room may have been particularly effective in supporting students who typically struggle in conventional learning settings. By embedding course content within an interactive game scenario that included points, nicknames, feedback, and leaderboards, the learning activities promoted motivation and knowledge retention. This finding is consistent with Welbers et al. [26], who demonstrated the potential of gamification to enhance student engagement in learning activities. Recent work by Mauri-Medrano et al. [27] also found that students exposed to escape room learning showed increased levels of fun and creative thinking that align closely with our observed results.

Aligned with the work of Lampropoulos and Kapoula [28] and Gerber and Fischetti [29], these findings reinforce the value of gamification in transforming passive content delivery into interactive and problem-solving experiences. In this study, the escape room tasks were designed to reflect core learning objectives from the database course, complemented by a short gamified learning session before the ER quizzes. Furthermore, students received immediate feedback during the gamified escape room activities and could identify errors in their responses. The paper-based quizzes did not provide this immediate feedback. This learning activity likely enhanced student motivation and knowledge retention, as reflected in the significantly higher post-quiz means from the gamified ER-based sessions (98.67 and 99.83) compared to the paper-based counterparts (80.58 and 85.17).

Moreover, the students' feedback obtained during Week 4 further supports the quantitative results. All students reported being very satisfied with the Escape Room activity and recommended it for future use. The preference of all students towards the gamified method rather than the paper-based quiz reflects its perceived effectiveness and demand in enhancing classroom learning activities.

Furthermore, the students reported high engagement across all UES dimensions. The elements in the gamified ER, such as leaderboards, nicknames, and points, likely increased students' motivation and active participation. This finding is consistent with Wiebe et al. [30], in which the work reported that higher engagement levels in game-based environments are associated with improved performance outcomes.

In summary, these outcomes support the use of escape rooms integrated with gamification concepts as an effective pedagogical tool. In addition to reinforcing conceptual understanding, this approach also fosters student motivation and encourages equitable participation, which are important contributors to academic success and sustained motivation in learning.

VI. CONCLUSION

This study demonstrates that gamified ER-based quiz activities can significantly enhance students' learning performance, motivation, and engagement in higher education, particularly in content-intensive subjects like database design. The within-subject design allowed students to experience both traditional and gamified approaches, and the results clearly showed higher post-quiz scores in the gamified sessions. The integration of game elements such as points, nicknames, and leaderboards provided an engaging, competitive, and immersive environment that supported knowledge retention and active learning. The user experience findings also indicate that students responded positively to both the practical and experiential aspects of the activity. High ratings for focused attention, perceived usability, aesthetic appeal, and reward are in line with the established four-factor structure of the UES, which has been shown to have good reliability and validity. Aesthetic appeal received the strongest evaluation, suggesting that visually appealing and interactive design features play an important role in sustaining engagement. This fits prior research showing that aesthetic appeal is one of the most stable engagement dimensions, while enjoyable and activating escape-room experiences are consistently associated with stronger student participation and positive learning perceptions. Moreover, the consistent preference for the gamified quizzes over traditional methods highlights the effectiveness of interactive learning designs in fostering student participation and a high level of engagement. These findings are consistent

with previous research, which supports the use of gamification as an effective pedagogical tool for enhancing student engagement. They align with the growing need to transform conventional assessments into dynamic learning experiences that encourage self-directed learning. Therefore, educators are encouraged to explore and implement gamified strategies like escape rooms to enrich instructional delivery, improve learning outcomes, and support 21st-century educational goals.

VII. Limitations and Future Work

This study was conducted among undergraduate students enrolled in a database course at a single institution, which limits the generalizability of the findings. Future research should involve multiple institutions and educational contexts to examine the broader applicability of the gamified escape room approach. Longer-term studies could assess its effects on knowledge retention and sustained student engagement. Future research could also examine the individual contributions of immediate feedback, points, and leaderboards to determine which elements most effectively support learning.

Acknowledgment

The authors acknowledge the UNIMAS Scholarship of Teaching and Learning Research Grant (UNI/F08/SoTL-RG/85796/2023) and Universiti Malaysia Sarawak for supporting this project. We also thank Neoh Jin Ming for developing the application and providing technical support.

References

[1] M. T. Chi and R. Wylie, "The ICAP framework: Linking cognitive engagement to active learning outcomes," Educational Psychologist, vol. 49, no. 4, pp. 219–243, 2014.

[2] R. Eppmann, M. Bekk, and K. Klein, "Gameful experience in gamification: Construction and validation of a gameful experience scale [GAMEX]," Journal of Interactive Marketing, vol. 43, no. 1, pp. 98–115, 2018.

[3] R. Van Roy and B. Zaman, "Need-supporting gamification in education: An assessment of motivational effects over time," Computers & Education, vol. 127, pp. 283–297, 2018.

[4] P. C. Chiu, M. J. Rupitan, K. K. Kuok, H. Sharbini, N. H. Borhan, and A. Yeo, "The effects of gamification in space exploration learning," Journal of Advanced Research in Applied Sciences and Engineering Technology, pp. 265–274, 2024.

[5] M. Huang, M. S. M. Saleh, and I. A. Zolkepli, "The moderating effect of environmental gamification on the relationship between social media marketing and consumer-brand engagement: A case study of Ant Forest Gen Z users," Heliyon, vol. 10, no. 4, 2024.

[6] T. Li, L. M. C. Castro, K. Douglas, and C. G. Brinton, "Relationship between learning engagement metrics and learning outcomes in online engineering course," in Proc. IEEE Frontiers in Education Conf. (FIE), Oct. 2021, pp. 1–5.

[7] M. Wang and X. Zheng, "Using game-based learning to support learning science: A study with middle school students," Asia-Pacific Educ. Res., vol. 30, no. 2, pp. 167–176, 2021, doi: 10.1007/s40299-020-00523-z.

[8] Z. Zhan, L. He, Y. Tong, X. Liang, S. Guo, and X. Lan, "The effectiveness of gamification in programming education: Evidence from a meta-analysis," Comput. Educ.: Artif. Intell., vol. 3, p. 100096, 2022, doi: 10.1016/j.caeai.2022.100096.

[9] P. Jomsri, D. Prangchumpol, and K. Poonsilp, "Enhancing student engagement in HCI courses through gamified mobile learning interfaces in higher education," International Journal of Information and Education Technology, vol. 15, no. 4, pp. 812–819, 2025, doi: 10.18178/ijiet.2025.15.4.2287.

[10] A. Aldabergenova, U. Yesseikyzy, I. Yessengabylov, A. Yesseikyzy, and A. Yerkinova, "Effectiveness of gamification elements in teaching programming to future informatics teachers," International Journal of Information and Education Technology, vol. 16, no. 3, pp. 601–610, 2026, doi: 10.18178/ijiet.2026.16.3.2532.

[11] J. Schildgen, "MonstER park - The entity-relationship-diagram learning game," CEUR Workshop Proceedings, vol. 2716, pp. 150–157, 2020.

[12] O. Dæhli, B. Kristoffersen, P. Lauvås, and T. Sandnes, "Exploring feedback and gamification in a data modeling learning tool," Electronic Journal of E-Learning, vol. 19, no. 6, pp. 559–574, 2021, doi: 10.34190/ejel.19.6.2501.

[13] W. Deng, Z. Pi, W. Lei, Q. Zhou, and W. Zhang, "Pencil Code improves learners' computational thinking and computer learning attitude," Computer Applications in Engineering Education, vol. 28, no. 1, pp. 90–104, 2020, doi: 10.1002/cae.22177.

[14] E. Barra, S. López-Pernas, A. Gordillo, A. Pozo, A. Munoz-Arcentales, and J. Conde, "Empowering database learning through remote educational escape rooms," IEEE Internet Computing, vol. 28, no. 1, pp. 18–25, 2023.

[15] B. M. Lunt et al., "Information Technology 2008: Curriculum guidelines for undergraduate degree programs in information technology," ACM and IEEE Comput. Soc., 2008, doi: 10.1145/362552.362554.

[16] ACM/IEEE Computer Society Task Group on Information Technology Curricula, Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology, 2017. [Online]. Available: https://www.acm.org/binaries/content/assets/education/it2017.pdf

[17] H. L. O'Brien, P. Cairns, and M. Hall, "A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form," International Journal of Human-Computer Studies, vol. 112, pp. 28–39, 2018.

[18] R. N. S. Amriza, K. N. M. Ngafidin, and C. Wiguna, "The examination of the User Engagement Scale (UES) in small medium enterprise social media usage: A survey-based quantitative study," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 6, pp. 1354–1362, 2023.

[19] C. Dichev and D. Dicheva, "Gamifying education: What is known, what is believed and what remains uncertain: A critical review," Int. J. Educ. Technol. High. Educ., vol. 14, pp. 1–36, 2017.

[20] S. Subhash and E. A. Cudney, "Gamified learning in higher education: A systematic review of the literature," Comput. Human Behav., vol. 87, pp. 192–206, 2018.

[21] A. Fuentes-Cabrera, M. E. Parra-González, J. López-Belmonte, and A. Segura-Robles, "Learning mathematics with emerging methodologies—The escape room as a case study," Mathematics, vol. 8, no. 9, p. 1586, 2020.

[22] Á. A. Magreñán, C. Jiménez, L. Orcos, and S. Roca, "Teaching calculus in the first year of an engineering degree using a digital escape room in an online scenario," Comput. Appl. Eng. Educ., vol. 31, no. 3, pp. 676–695, 2023.

[23] D. Miranda, C. Li, and T. Darin, "Ues-br: Translation and cross-cultural adaptation of the user engagement scale for brazilian portuguese," Proceedings of the ACM on Human-Computer Interaction, vol. 5, CHI PLAY, pp. 1–22, 2021.

[24] Z. Zainuddin, M. Shujahat, H. Haruna, and S. K. W. Chu, "The role of gamified e-quizzes on student learning and engagement: An interactive gamification solution for a formative assessment system," Comput. Educ., vol. 145, p. 103729, 2020.

[25] A. Gordillo and D. López-Fernández, "Are educational escape rooms more effective than traditional lectures for teaching software engineering? A randomized controlled trial," IEEE Trans. Educ., 2024.

[26] K. Welbers, E. A. Konijn, C. Burgers, A. B. de Vaate, A. Eden, and B. C. Brugman, "Gamification as a tool for engaging student learning: A field experiment with a gamified app," E-Learning Digit. Media, vol. 16, no. 2, pp. 92–109, 2019, doi: 10.1177/2042753018818342.

[27] M. Mauri-Medrano, J. L. Giménez-López, and J. Castejón-Costa, "Comparing an educational escape room with a digital board game: Effects on fun, creative thinking, and mastery," Front. Educ., vol. 9, p. 1354674, 2024, doi: 10.3389/feduc.2024.1354674.

[28] G. Lampropoulos and A. Sidiropoulos, "Impact of gamification on students' learning outcomes and academic performance: A longitudinal study comparing online, traditional, and gamified learning," Educ. Sci., vol. 14, no. 4, p. 367, 2024.

[29] A. Gerber and B. Fischetti, "The impact of escape room gamification using a teleconferencing platform on pharmacy student learning," Med. Sci. Educ., vol. 32, no. 5, pp. 1159–1164, 2022.

[30] E. N. Wiebe, A. Lamb, M. Hardy, and D. Sharek, "Measuring engagement in video game-based environments: Investigation of the User Engagement Scale," Computers *in Human Behavior*, vol. 32, pp. 123–132, 2014, doi: 10.1016/j.chb.2013.12.001.

APPENDIX

TABLE VI. UES ITEMS

Item	UES
AE.1	This Application X was attractive.
AE.2	This Application X was aesthetically appealing.
AE.3	I liked the graphics and images of Application X.
AE.4	Application X appealed to my visual senses.
AE.5	The screen layout of Application X was visually pleasing.
FA.1	I lost myself (deeply focused) in this experience.
FA.2	I enjoyed the experience of using Application X that I lost track of time.
FA.3	I blocked out activities around me when I was using Application X.
FA.4	When I was using Application X, I lost track of the world around me.
FA.5	The time I spent using Application X just slipped away.
FA.6	I was absorbed (interested) in this experience.
FA.7	During this experience I let myself enjoy it.
PU.1	I felt frustrated while using this Application X.
PU.2	I found this Application X confusing to use.
PU.3	I felt annoyed while using Application X.
PU.4	I felt discouraged while using this Application X.
PU.8	I could not do some of the things I needed to do while using Application X.
PU.5	Using this Application X was tiring.
PU.6	This experience of Application X browsing is troublesome.
PU.7	I felt in control while using this Application X.
RW.1	Using Application X was worthwhile.
RW.2	I consider my experience a success.
RW.3	This experience did not work out the way I had planned.
RW.4	My experience was rewarding.
RW.5	I would recommend Application X to my peers and friends.
RW.6	I continued to use Application X out of curiosity.
RW.7	The content of Application X sparked my interest.
RW.8	I am really interested into this experience.
RW.9	I felt involved in this experience.
RW.10	This experience was fun.

Fall Detection System Using Accelerometer and Gyroscope Sensors with Backpropagation Neural Networks

Helmy Fitriawan[1]*, Alby Dzaky Hadi Pamungkas[2], Ezza Ahmad Faturohman[3],
Sri Purwiyanti[4], Anisa Ulya Darajat[5], Teddy Surya Gunawan[6]
Dept. of Electrical Engineering, University of Lampung, 35145 Bandar Lampung, Indonesia[1, 2, 3, 4, 5]
Electrical and Computer Engineering Dept., International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia[6]

***Abstract*—A fall is an incident in which a person suddenly loses their balance and falls uncontrollably to the ground or another surface. This is particularly dangerous for the elderly, as it can lead to serious injury or even death if not treated immediately. This study suggests utilizing a Backpropagation Neural Network (BPNN) to classify fall and non-fall incidents, particularly in older adults, by measuring alterations in linear acceleration and angular velocity obtained from accelerometer and gyroscope sensors. A dataset featuring eight distinct movement types, including standing to bowing, standing to squatting, sitting to standing, standing to prostrating, sitting to prostrating, falling sideways, falling forward, and falling backward, is collected from 69 participants and processed by determining the difference (Δ) between consecutive sensor measurements, followed by normalization, labeling, and statistical feature extraction. The collected features are fed into a BPNN that has two hidden layers, each with 64 and 128 neurons. The proposed model achieved an accuracy of 86.87%, validation accuracy of 85.00%, and testing accuracy of 92.86%. Furthermore, test results show a precision of 94%, recall of 85%, and specificity of 97%, demonstrating the system's ability to accurately classify eight different human activities and distinguish falls from normal daily activities. These findings indicate that a Backpropagation Artificial Neural Network in conjunction with the suggested Δ-based preprocessing offers an efficient method for inertial sensor-based fall detection systems.**

***Keywords*—Backpropagation neural network; fall detection; human activity recognition; accelerometer; gyroscope**

I. Introduction

Patient safety is of utmost importance, and falls are among the most frequent incidents affecting elderly adults. For elderly patients, experiencing unnoticed falls can pose a hidden danger. The immediate consequences of a fall, such as injury, can be worse if assistance is not obtained immediately. Statistics indicate that the most serious outcomes result from delays in receiving help and medical care, rather than from the fall itself [1-4].

In the healthcare sector, the development of fall detection devices has become a prominent research topic. Recent advancements in Micro-Electro-Mechanical Systems (MEMS) sensor and actuator technology have enabled the development of fall detection systems using accelerometers and gyroscopes [5-8]. This technique relies on detecting and tracking changes in acceleration along three orthogonal directions to identify changes in a person's body position and movement. A threshold-based fall detection system uses wearable sensors to detect sudden body movements. The system monitors when total acceleration exceeds a pre-set numerical threshold value. If the data surpasses that limit, the system flags a potential fall event [9-12]. The collected data is continuously analyzed algorithmically to determine whether the individual has fallen. If a fall is detected, the device can use GPS and a wireless transmitter to determine the location and issue an alert for help. Research has also been conducted on fall detection using smartphone accelerometers, where the system can send an SMS containing an alert message indicating that the patient has fallen, along with the person's coordinates, and automatically connect the call to a pre-registered number [13-16]. The core elements of fall detection are reliable detection principles and practical algorithms for assessing the existence of emergencies.

Various classification methods have been applied to improve the accuracy of fall detection. Research using the Fuzzy Logic method has been able to detect falls based on a rule base constructed from changes in acceleration values and body orientation. However, the performance of this method is highly dependent on the formulation of rules and membership functions, making it less flexible in recognizing complex movement variations [17-20]. Another study using the K-Nearest Neighbor (KNN) algorithm showed that accelerometer and gyroscope data can be classified with a good level of accuracy, but its performance is influenced by the choice of the K value, the distribution of the training data, and increased computation time as the dataset size grows [21-23].

Addressing these issues, this study develops an activity classification and fall detection model using a Backpropagation Neural Network that utilizes accelerometer and gyroscope data from the MPU6050 sensor. This study employs a relatively large dataset—comprising 36,432 data points collected from 69 participants—covering eight types of movements: standing to bowing, standing to squatting, sitting to standing, standing to prostrating, sitting to prostrating, falling sideways, falling forward, and falling backward. Furthermore, the study implements preprocessing stages—including the calculation of inter-sample value changes (Δ), normalization, and data labeling—prior to the training process to generate more representative features. The primary contribution of this

*Corresponding author.
Higher Education for Technology and Innovation (HETI) Project.

research is the development of a Backpropagation Neural Network model that utilizes a larger dataset and comprehensive preprocessing steps to enhance the classification performance for these eight movement types based on MPU6050 sensor data. Model performance is evaluated using a confusion matrix based on accuracy, precision, recall, and specificity metrics to assess the ability of the model to accurately recognize activities and detect falls [24-26].

The following sections of this study are organized as follows: Section II presents the stages of this research and proposes the backpropagation neural network architecture employed. Model training and the discussion of test results are presented in Section III. Finally, research findings are concluded in Section IV.

II. METHOD

This study aims to develop a fall detection system utilizing accelerometer and gyroscope sensors integrated into an MPU6050 module. An Artificial Neural Network (ANN) employing the backpropagation method is used to process sensor data and identify various movement patterns, thereby distinguishing between falls and normal daily activities.

Fig. 1 shows the workflow of the research methodology. In the initial phase of the research, the MPU6050 module, comprising accelerometer and gyroscope sensors mounted on a wearable vest, is used to collect data. These sensors continuously recorded movement data from subjects performing various movement scenarios, including simulated falls and daily activities. The collected raw signals subsequently underwent preprocessing to remove noise and normalize the data, ensuring uniformity for further analysis.

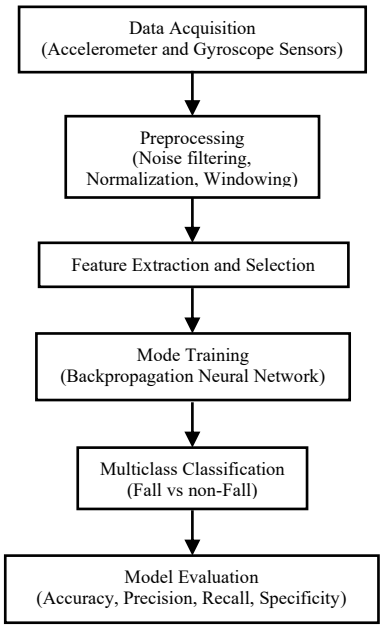


Fig. 1. Research stages of a fall detection system using a Backpropagation Neural Network (BPNN).

Following preprocessing, pertinent movement characteristics that characterize various movement patterns are obtained using a feature extraction and selection process. A BPNN model is then trained using the backpropagation algorithm utilizing the chosen features as input. The data set is split into training and testing subsets in order to assess the model performance and guarantee cross-validation of the classification procedure. The BPNN architecture is optimized by varying the number of hidden layers and neurons to achieve the highest accuracy. Model performance is evaluated based on accuracy, precision, recall, and specificity.

The following part provides a detailed description of how the proposed fall detection system is implemented.

A. Data Collection

The dataset in this study is collected using primary data acquisition from subjects wearing a vest equipped with the MPU6050 sensor module, which combines a 3-axis accelerometer and a 3-axis gyroscope. This sensor is placed on the user's shoulder and operated via an Arduino-based system. This placement is chosen to capture movement data and changes in body orientation with greater accuracy, given that the shoulder area is relatively stable and less prone to the extraneous movements often associated with the arms or legs. Thanks to this consistent and stable positioning, the sensor can measure angular changes and body movements with a higher degree of precision. Furthermore, the shoulder surface, which tends to remain relatively parallel to the earth, helps enhance measurement accuracy while minimizing interference caused by unexpected movements.

The dataset consists of motion signals recorded from simulated fall events as well as normal daily activities. In this research, the dataset was collected from 69 participants, where data from 62 participants are used for model training and data from the remaining 7 participants are used for model testing. Each participant took part in a data collection session covering two types of movements: non-fall movements (standing to bowing, standing to squatting, sitting to standing, standing to prostrating, and sitting to prostrating) and fall movements (falling sideways, falling forward, and falling backward). Conducting fall experiments on older adults entails certain risks, even under controlled settings; therefore, the fall simulations for this study were conducted exclusively and in a controlled manner on young participants between the ages of 20 and 24.

In order to gain coherent movement patterns, each activity is recorded from the initial position to the final position of the movement. Specifically for simulated fall activities, the recording process begins when the subject is in an upright position, continues as the subject loses balance, and concludes when the subject reaches the final position after the fall. This series of stages is carried out to fully represent the fall movement pattern, which will subsequently be used as input data for the BPNN. An illustration of the data acquisition process when a participant experiences a fall is shown in Fig. 2. The complete dataset is then split into training and testing subsets to evaluate the model performance.



Fig. 2. Illustration of data collection from a subject who has experienced a fall.

B. Preprocessing and Feature Selection

Raw data obtained from sensors undergoes a preprocessing stage to enhance its quality and consistency. These preprocessing steps include noise filtering, normalization, and the segmentation of time-series signals into windows suitable for feature extraction. This stage ensures uniform data scaling and minimizes measurement errors, rendering the data more suitable for analysis using an ANN. Data is measured and recorded using an MPU6050 sensor, which yields six parameters: acceleration (Ax, Ay, and Az) and angular velocity (Gx, Gy, and Gz) along the x, y, and z axes, respectively. Feature extraction and selection are carried out to identify motion characteristics most relevant to fall detection. Each activity is recorded over 11 samples, so that each movement produced an 11 x 6 data matrix representing changes in acceleration and body orientation throughout the activity, as presented in Table I.

TABLE I. RAW DATA FROM ACCELEROMETER AND GYROSCOPE SENSOR READINGS BEFORE PREPROCESSING

No	Accelerometer (A)			Gyroscope (G)		
	Ax	Ay	Az	Gx	Gy	Gz
1	-1.60	-3.70	8.52	0.09	0.09	-0.24
2	-1.66	-3.71	8.51	0.64	0.49	-1.19
3	-1.53	-3.76	8.49	0.99	0.05	-4.84
4	-1.47	-3.70	8.53	-0.17	-0.40	0.40
5	-1.53	-3.71	8.53	-1.59	-3.97	-0.96
6	-1.79	-3.60	7.29	-19.73	-57.05	-12.08
7	6.93	-4.36	3.65	-24.90	-103.53	-35.62
8	9.94	-4.41	-1.97	-5.69	-53.59	-24.87
9	9.79	-3.29	-3.38	0	-0.95	-7.31
10	8.75	-3.52	-2.78	4.18	-0.12	-1.37
11	9.05	-3.32	-2.91	1.98	-0.98	-1.34

Before being used as input for the BPNN model, the data undergoes preprocessing steps that include dataset preparation, calculation of the change in values between samples (Δ), normalization, and data labeling. After the raw data is obtained, the next step is to calculate the difference (Δ) between samples for each parameter of the accelerometer (Ax, Ay, and Az) and the gyroscope (Gx, Gy, and Gz). This calculation aims to represent the changes in acceleration and angular velocity that occur during the activity, so that movement characteristics can be captured more accurately than when using raw sensor values. Additionally, using the Δ values helps reduce the influence of differences in the initial sensor positions among participants. Since each activity consists of 11 samples, the Δ calculation process yields 10 data points for each sensor parameter, which are then used as input features in the Backpropagation Neural Network model as presented in Table II.

Based on the calculated change values (Δ), each data point is then labelled according to the type of activity performed. The labelling process aims to distinguish eight movement classes: standing to bowing, standing to squatting, sitting to standing, standing to prostrating, sitting to prostrating, falling sideways, falling forward, and falling backward. The data that has undergone the change value calculation and labelling processes is then used as input for the feature extraction stage and the training of the BPNN.

TABLE II. SAMPLE OF DIFFERENCE (Δ) VALUES OBTAINED FROM ACCELEROMETER AND GYROSCOPE DATA

No	Accelerometer (A)			Gyroscope (G)		
	ΔAx	ΔAy	ΔAz	ΔGx	ΔGy	ΔGz
1	-0.06	-0.01	-0.01	0.55	0.40	-0.95
2	0.13	-0.05	-0.02	0.35	-0.44	-3.65
3	0.06	0.06	0.04	-1.16	-0.45	5.24
4	-0.06	-0.01	0	-1.42	-3.57	-1.36
5	-0.26	0.11	-1.24	-18.14	-53.08	-11.12
6	8.72	-0.76	-3.64	-5.17	-46.48	-23.54
7	3.01	-0.05	-5.62	19.21	49.94	10.75
8	-0.15	1.12	-1.41	5.69	52.64	17.56
9	-1.04	-0.23	0.60	4.18	0.83	5.94
10	0.30	0.20	-0.13	-2.20	-0.86	0.03

C. Modelling

In the modelling phase, the classification process is carried out using an ANN trained with the backpropagation algorithm. The model training process is implemented using the Python programming language, which offers flexibility in optimizing network parameters. Before the training process begins, each parameter representing the change (Δ) is transformed into statistical features consisting of the minimum (Min), maximum (Max), mean (Mean), and standard deviation (Standard Deviation) values [27-29]. This feature extraction process summarizes the key characteristics of each activity, reducing data complexity while preserving essential information that represents movement patterns. The minimum and maximum values represent the lowest and highest sensor readings during the activity, respectively; the mean value describes the trend of sensor changes, while the standard deviation measures data variation relative to that mean. These four statistical features are calculated for each accelerometer axis (Ax, Ay, Az) and gyroscope axis (Gx, Gy, Gz), resulting in a total of 24 input features.

The extracted features are then used to train the Backpropagation Neural Network. The neural network architecture consists of one input layer, two hidden layers, and one output layer. The hidden layers process the extracted features, while the output layer generates binary or multi-class

classification results depending on the scenario. The architecture of the proposed BPNN model employed in this study is illustrated in Fig. 3. The input layer contains 24 neurons (i_1, i_2, …, i_{24}) corresponding to the extracted statistical features. The first and second hidden layers consist of 128 ($h_{1,1}$, $h_{1,2}$, …, $h_{1,128}$) and 64 ($h_{2,1}$, $h_{2,2}$, …, $h_{2,64}$) neurons, respectively, while the output layer consists of 8 neurons (o_1, o2, …, o_8) representing the eight observed activity classes.

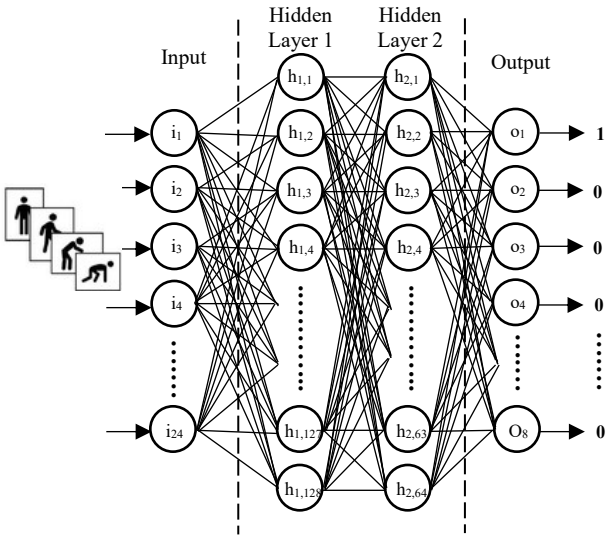


Fig. 3. Architecture of the backpropagation neural network employed in this study.

The input layer receives 24 statistical features derived from accelerometer and gyroscope data obtained during the pre-processing and feature selection stages. The output layer has 8 neurons representing eight activity classes: class 0 for standing to bowing, class 1 for standing to squatting, class 2 for sitting to standing, class 3 for standing to prostrating, class 4 for sitting to prostrating, class 5 for falling sideways, class 6 for falling forward, and class 7 for falling backward.

The proposed network incorporates two hidden layers because the accelerometer and gyroscope data exhibit complex non-linear characteristics. A narrowing structure (from 128 neurons to 64 neurons) was selected for this network to mimic the principle of gradual feature abstraction. The first hidden layer (128 neurons) serves to learn basic patterns and raw features from the input, while the second hidden layer (64 neurons) captures more complex relationships between features and distills that information into a more concise and meaningful representation, thereby enabling more optimal classification at the output layer [30]. Research in [31] indicates that the 128-64 neuron configuration achieves an optimal point (sweet spot) on the performance curve. This configuration yields high accuracy without showing signs of overfitting, thereby maintaining the model's flexibility when processing new data.

Both hidden layers use the ReLU activation function to handle non-linear decision boundaries. These functions satisfy the properties of continuity, differentiability, and monotonicity, which are essential for the learning process in artificial neural networks. It also accelerates the convergence during the network training process and prevents the vanishing gradient problem often encountered in artificial neural network models [32-33]. The ReLU (Rectified Linear Unit) activation function transforms negative input values into zero and passes positive values through directly, using Eq. (1).

$$ReLU(z) = max(0, z) \quad (1)$$

where, z is the neuron input value; if the input value is below zero ($z < 0$), the output is 0, and if the input value is greater than or equal to zero ($z \geqslant 0$), the output remains equal to the input value itself.

The output layer employs the Softmax activation function to transform the network channels into a probability distribution across the eight activity classes. Each probability value lies between 0 and 1, and the total sum of values at the output layer is always exactly 1 [34-35]. The Softmax function is defined in Eq. (2).

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad (2)$$

where, $\hat{y}_i$ is the probability of the ith class or the actual true label, z is output of the neuron before activation, e^{z_i} is the exponential function, n is the number of output classes, and $\sum_{j=1}^{n} e^{z_j}$ is the sum of all exponential values. The class with the highest probability value is the network prediction.

The training phase utilizes a training dataset and involves forward propagation, loss calculation, and bias and weight updates using the backpropagation algorithm. This process yields accuracy and loss metrics that indicate the quality of the training. The final weights and biases obtained from the training process are subsequently used to classify activities based on the testing dataset.

Forward propagation processes input data in a single forward direction through the layers of a neural network to generate a final prediction. Each neuron within a layer processes incoming values, x, using specific weights, w, and biases, b, transforms them via an activation function, and then passes the result, z, to the next layer, as shown in Eq. (3).

$$z = x \cdot w + b \quad (3)$$

After the forward propagation process generates a prediction at the output layer, the loss value is calculated to measure the magnitude of the difference between the network prediction and the actual ground-truth label. In our proposed BPNN architecture, the categorical cross-entropy loss function is used as it is suitable for multiclass classification problems, specifically the eight-class classification as discussed here. It measures the dissimilarity between the predicted probability distribution and the true one-hot encoded label distribution. The lower the resulting loss value, the better the model's ability to predict the correct class. The equation for the categorical cross-entropy loss function is shown in Eq. (4).

$$L = -\sum_i^n y_i \log(\hat{y}_i) \quad (4)$$

Once the loss value is obtained, the backpropagation process is performed to calculate the error gradients with respect to each weight and bias in the network. These gradients are used to iteratively update the network parameters so that the loss value decreases and the model's ability to classify activities improves.

Eq. (5) shows the weight update in the backpropagation algorithm using the network learning rate, denoted byη.

$$W_{new} = W_{old} - \eta \cdot \frac{dL}{dW} \quad (5)$$

A similar equation is also applied to the update in bias values.

D. Evaluation

The evaluation of the proposed fall detection system is conducted using a confusion matrix, which provides a detailed representation of classification results by distinguishing between correctly and incorrectly predicted classes. A multiclass confusion matrix for a neural network is an N × N grid (where N is the number of classes) that contrasts the network-predicted labels (columns) with the true target labels (rows). The diagonal cells display accurate predictions, while off-diagonal cells reveal particular misclassifications. The confusion matrix consists of four components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These values enable the calculation of various performance metrics, including accuracy, precision, recall, and specificity, which are determined using the following equations.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$Specificity = \frac{TN}{TN+FP} \quad (9)$$

Accuracy is the total correct predictions (sum of the diagonal) divided by the total samples; precision is true positives divided by all positive predictions, recall is true positives divided by the total actual positives; and specificity measures the proportion of actual negative cases correctly identified as negative by the neural network.

From the optimization perspective, the loss function guides network training, while these evaluation metrics evaluate real-world practical performance. In this study context, they provide a thorough review of an ANN-based fall detection system while guaranteeing classification accuracy and dependability.

III. Results and Discussion

This section presents the results of the proposed BPNN training process using the preprocessed training dataset, as well as the testing results using the test dataset. Afterward, the performance of the model is evaluated using metrics derived from the confusion matrix.

A. Model Training

A neural network learns patterns from data by modifying its internal biases and weights during training. The training process is evaluated based on changes in accuracy and loss values at each epoch to assess the model ability to learn movement patterns and its convergence rate. The variation in accuracy values throughout the training process is depicted in Fig. 4. At the beginning of the training, the accuracy level is low because the model was not yet able to optimally recognize data patterns. As the number of epochs increased, accuracy improved significantly, eventually reaching approximately 86% by the end of the training. The curve, which begins to flatten in subsequent epochs, indicates that the model has achieved convergence and is capable of effectively learning the characteristics of the training dataset of movement patterns.

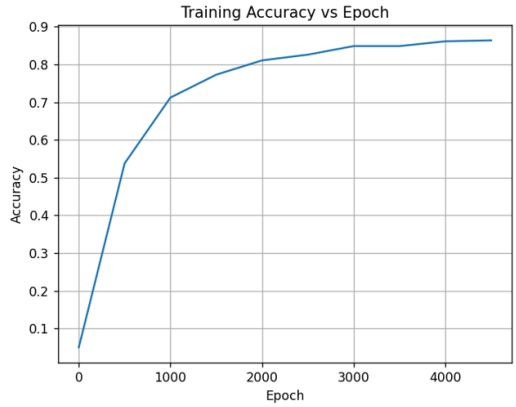


Fig. 4. Plot of accuracy vs epoch for training data.

The loss value indicates the magnitude of the prediction error produced by the model during the training process. The smaller the loss value, the better the model's ability to learn data patterns. Fig. 5 shows that the loss value decreases significantly at the beginning of the training process. As the number of epochs increases, the loss value continues to decrease until it reaches a relatively stable state. This decrease indicates that the optimization process has successfully minimized prediction errors and the model has reached a state of convergence.

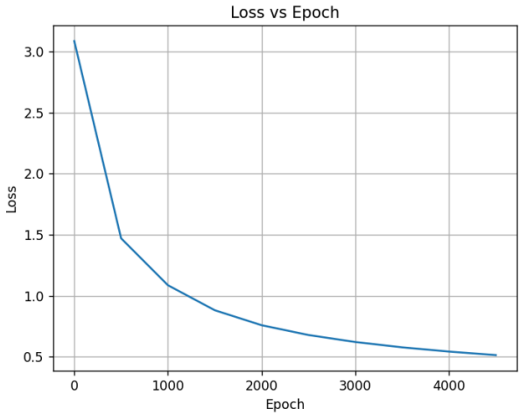


Fig. 5. Plot of loss vs epoch for training data.

B. Classification Performance

After the training process is completed, the proposed BPNN model is evaluated using test data to determine its ability to classify eight types of activities based on accelerometer and gyroscope sensor data. The evaluation is conducted using a confusion matrix, which provides information on the number of correct and incorrect predictions for each activity class. Based on this confusion matrix, the accuracy, precision, recall, and specificity are then calculated to measure the model's overall performance, as shown in Fig. 6.

Actual Data \ Prediction	Standing to Bowing (0)	Standing to Squatting (1)	Sitting to Standing (2)	Standing to Prostrating (3)	Sitting to Prostrating (4)	Falling Sideways (5)	Falling Forward (6)	Falling Backward (7)
Standing to Bowing (0)	7	0	0	0	0	0	0	0
Standing to Squatting (1)	1	6	0	0	0	0	0	0
Sitting to Standing (2)	0	0	7	0	0	0	0	0
Standing to Prostrating (3)	0	0	0	7	0	0	0	0
Sitting to Prostrating (4)	0	0	0	0	7	0	0	0
Falling Sideways (5)	0	0	0	0	0	6	0	1
Falling Forward (6)	0	0	0	2	0	0	5	0
Falling Backward (7)	0	0	0	0	0	0	0	7

Fig. 6. Confusion matrix of the BPNN to classify eight different movement patterns, including fall and non-fall.

The model is tested using 56 test data points consisting of seven participants across eight movement pattern classes. The test results show that 52 data points are classified correctly, while 4 data points are misclassified. These errors consisted of one sample of standing to squatting that is predicted as standing to bowing, two samples of falling forward that are predicted as standing to prostrating, and one sample of falling sideways that is predicted as falling backward. Test results revealed three movement patterns that were incorrectly predicted due to striking similarities among them; for instance, the transition from a standing to a squatting position closely resembles the movement from standing to bending over, while falling forward is similar to moving from standing to a prostrate position, and finally, falling sideways closely resembles falling backward. These classification errors indicate that some activities have similar patterns of acceleration and angular velocity changes, making it difficult for the model to distinguish between certain classes. Nevertheless, these results demonstrate that the model is capable of classifying the eight types of activities effectively.

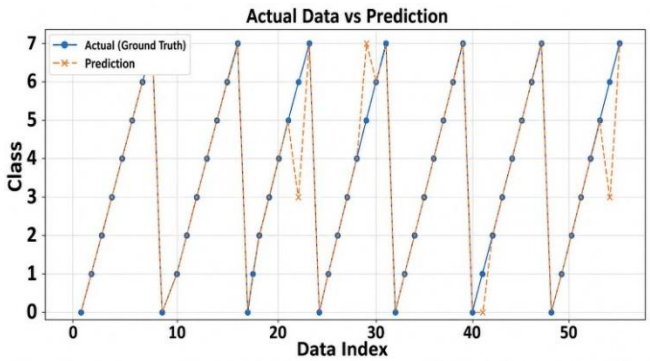


Fig. 7. Actual vs predicted class from all test data.

A comparison is made between the actual labels and the anticipated findings from all test data in order to produce a more comprehensible picture of the categorization results, as shown in Fig. 7. The model prediction results show all patterns nearly identical to the actual data. Most of the predicted points coincide with the actual data, indicating that the model is capable of accurately recognizing the patterns of the eight activity types. Differences are only visible at a few points representing the four samples that are misclassified, as explained in the confusion matrix results. This indicates that the model performs well in predicting activities based on accelerometer and gyroscope sensor data.

The classification results are then assessed using evaluation metrics, such as accuracy, precision, recall, and specificity, as described in Eq. (6)- (9), to obtain a quantitative estimate of the model performance. Based on the results of the confusion matrix, the model achieved an accuracy of 92.86%, indicating that the majority of the test data is classified correctly. Testing accuracy is higher than training accuracy; this is likely because the test data is smaller, cleaner, or inherently simpler compared to the training dataset. The test data comprises approximately 10% of the total dataset, whereas the training data accounts for 90%. In addition, the model achieved a precision of 94%, a recall of 85%, and a specificity of 97%. These values indicate that the model performs well in recognizing the eight types of activities and is able to accurately distinguish between falls and normal activities.

IV. Conclusion

This research successfully developed a comprehensive architecture and algorithm to detect fall and non-fall movement patterns using the backpropagation method within an artificial neural network. The designed system is capable of training on and predicting movement patterns based on data obtained from accelerometer and gyroscope sensors, thereby demonstrating its practical application in real-world scenarios.

Prior to the training process, the raw sensor data are preprocessed by calculating the difference (Δ) between consecutive samples and assigning activity labels to generate more representative input features. The dataset was collected from 69 participants, where data from 62 participants are used for model training and data from the remaining 7 participants are used for model testing. The training process achieved a training accuracy of 86.87% and a validation accuracy of 85.00%, indicating that the proposed model is able to learn the activity patterns effectively while maintaining good generalization performance. The proposed model is evaluated using a confusion matrix on the testing dataset. Experimental results showed that the model correctly classified 52 out of 56 activity samples, achieving an accuracy of 92.86%, precision of 94%, recall of 85%, and specificity of 97%. These findings demonstrate that the proposed backpropagation neural network, combined with the Δ-based preprocessing approach, is capable of accurately classifying eight human movement patterns and has the potential to be applied in inertial sensor-based fall detection system.

Acknowledgment

The Higher Education for Technology and Innovation (HETI) program at the University of Lampung is acknowledged by the author for providing funds for this study.

References

[1] A. Bueno-Cavanilas, F. Padilla-Ruiz, J. J. Jimenez-Moleon, C. A. Peinado-Alonso, and R. Gavez-Vargaz, "Risk factors in falls among the elderly according to extrinsic and intrinsic precipitating causes," European Journal of Epidemiology, vol. 16, no. 9, pp. 849-859, Sep. 2000, doi: 10.1023/a:1007636531965.

[2] A. Bergland and T. B. Wyller, “Risk factors for serious fall-related injury in elderly women living at home,” Injury Perevention, vol. 10, no. 5, pp. 308-311, Oct. 2004, doi: 10.1136/ip.2003.004721.

[3] M. T. Aschkenasy and T. C. Rothenhaus, “Trauma and falls in the elderly,” Emergency Medicine Clinics of North America, vol. 24, no. 2, pp. 413–432, May 2006, doi: 10.1016/j.emc.2006.01.005

[4] Montero-Odasso M, et al., “ World guidelines for falls prevention and management for older adults: a global initiative,” Age Ageing, vol. 51, no. 9, pp. 1-36, Sep 2022, doi: 10.1093/ageing/afac205.

[5] F. Khoshnoud and C. W. de Silva, "Recent advances in MEMS sensor technology-mechanical applications," in IEEE Instrumentation & Measurement Magazine, vol. 15, no. 2, pp. 14-24, April 2012, doi: 10.1109/MIM.2012.6174574.

[6] E. Oporto, L.A.Vilcahuaman, “Fall detection system for elderly by MEMS accelerometer and SMS alert,” in 2016 VII Latin American Congress on Biomedical Engineering (CLAIB 2016), Bucaramanga, Santander, Colombia, Oct. 2016, pp.114-117, doi: 10.1007/978-981-10-4086-3_29.

[7] B. Yang, H. Xia, and W. Du, “A detection system for the elderly based on embedded systems,” in 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC 2020), Zhanjiang, China, Oct. 2020, pp. 505-508, doi: 10.1109/YAC51587.2020.9337589.

[8] L. Zheng, J. Zhao, F. Dong, Z. Huang, and D. Zhong, “Fall detection algorithm based on inertial sensor and hierarchical decision,” Sensor, vol. 23, no. 1, pp. 107/1-107/20, Dec. 2022, doi: 10.3390/s23010107.

[9] S. S. Jun, H. R. Ramli, A. C. Soh, N. A. Kamsani, R. K. R. Ahmad, S. A. Ahmad, and A. J. Ishak, “Development of fall detection and activity recognition using threshold based method and neural network,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, no. 3, pp. 1338–1347, Mar. 2020, doi: 10.11591/ijeecs.v17.i3.pp1338-1347.

[10] T. Xu, H. Se, and J. Liu, “A fusion fall detection algorithm combining threshold-based method and convolutional neural network,” Microprocessors and Microsystems, vol. 82, pp. 103828-103837, April 2021, doi: 10.1016/j.micpro.2021.103828.

[11] S. S. Jeong, N. H. Kim, and Y. S. Yu, “Fall detection system based on simple threshold method and long short-term memory: comparison with hidden markov model and extraction of optimal parameters,” Applied Sciences, vol. 12, no. 21, pp. 11031-11048, Oct. 2022, doi: 10.3390/app122111031.

[12] J. Fernandez-Bermejo, J. Martinez-del-Rincon, J. Dorado, X. del Toro, M. J. Santofimia, and J. C. Lopez, “Edge computing transformers for fall detection in older adults,” International Journal of Neural Systems, vol. 34, no. 5, pp. 2450026/1-2450026/20, May 2024, doi: 10.1142/S0129065724500266.

[13] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, “Development of a wearable-sensor-based fall detection system,” International Journal of Telemedicine and Applications, vol. 2015, pp. 576364/1-576364/11, 2015, doi: 10.1155/2015/576364.

[14] M. N. Farhan, M. G. Ayoub, A. R. H. Al-Jader, “ GPS-based fall detection system for old and specially-abled people,“ Indonesian Journal of Electrical Engineering and Computer Science, vol. 31, no. 3, pp. 1545-1550, Sep. 2023, doi: 10.11591/ijeecs.v31.i3.pp1545-1550.

[15] H. Fitriawan, S. Purwiyanti, E. A. Faturrohman, M. F. Santoso, A. U. Darajat, and T. S. Gunawan, “Development of a low-cost fall detection system for the elderly with accurate detection and real-time alerts,” in 2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA 2024), Bandung, Indonesia, July 2024, pp. 309–314, doi: 10.1109/ICSIMA62563.2024.10675589.

[16] C.-K. Tseng, S.-J.Huang, and L.-J. Kau, “Wearable fall detection system with real-time localization and notification capabilities,” Sensors, vol. 25, no. 12, pp. 3632/1-3632/134, Jun. 2025, https://doi.org/10.3390/s25123632.

[17] A. Ibrahim and M. I. Younis, “Simple fall criteria for MEMS sensors: data analysis and sensor concept,” Sensors, vol. 14, no. 7, pp. 12149-12173, July 2014, doi:10.3390/s140712149.

[18] P. V. Er and K. K. Tan, “Non-intrusive fall detection monitoring for the elderly based on fuzzy logic,” Measurement, vol. 124, pp. 91-102, Aug. 2018, doi: 10.1016/j.measurement.2018.04.009.

[19] H. Zhang, M. Alrifaai, K. Zhou, and H. Hu, “A novel fuzzy logic algorithm for accurate fall detection of smart wristband,” Transactions of the Institute of Measurement and Control, vol. 42, no. 4, pp. 786-794, Feb. 2020, doi: 10.1177/0142331219881578.

[20] M. A. Syamlan et al., "Fuzzy logic-based fall detection system for elderly using a single inertial measurement unit," in 2024 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM 2024), Surabaya, Indonesia, Nov. 2024, pp. 1-5, doi: 10.1109/CENIM64038.2024.10882737.

[21] C. Liu, C. Lee, and P. Lin, “A fall detection system using k-nearest neighbor classifier,” Experts Systems with Applications, vol. 37, no. 10, pp. 7174-7181, Oct. 2010, doi: 10.1016/j.eswa.2010.04.014.

[22] P. Bilski, P. Mazurek, J. Wagner, “Application of k nearest neighbors approach to the fall detection of elderly people using depth-based sensors,” in The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2015), Warsaw, Poland, Sep. 2015, pp. 722-729, doi: 10.1109/IDAACS.2015.7341400.

[23] D. Congzhang , Z. Ding, L. Wang, and Y. Jia, “A Fall Detection Method Based on K-Nearest Neighbor Algrithm with MIMO Millimeter-Wave Radar,” in 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP 2021), Nanjing, China, Oct. 2021, pp. 624-628 , doi: 10.1109/ICSIP52628.2021.9688752.

[24] A. Tharwat A, “Classification assessment methods,” Applied Computing and Informatics, vol. 17, no. 1, pp. 168–192, Jul. 2020, doi: 10.1016/j.aci.2018.08.003.

[25] J. E. Dobson, “On the confusion matrix,” Configurations, vol. 32, no. 4, Sep. 2024, pp. 331-350, doi: 10.1353/con.2024.a942087.

[26] S. Sathyanarayanan and B. R. Tantri, “Confusion matrix-based performance evaluation metrics,” African Journal of Biomedical Research, vol. 27, no. 4s, pp. 4023-4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.

[27] Giovanni Di Franco and Michele Santurro, “Machine learning, artifcial neural network and social research,” Quality & Quantity, vol. 55, no. 3, pp. 1007-1025, Jun. 2021, doi: 10.1007/s11135-020-01037-y.

[28] Y-S. Kim, M. K. Kim, N. Fu, J. Liu, J. Wang, J. Srebric, “Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models,” Sustainable Cities and Society, vol. 118, Jan. 2025, pp. 105570/1-105570/12, doi: 10.1016/j.scs.2024.105570.

[29] N. Hoang and V. Tran, “A data-driven approach based on deep neural network regression for predicting the compressive strength of steel fiber reinforced concrete,” Journal of Soft Computing in Civil Engineering, vol. 9, no. 2, pp. 1-31, Spring 2025, doi: 10.22115/scce.2024.430215.1765.

[30] E. Paluzo-Hidalgo, R. Gonzalez-Diaz, and M. A. Gutiérrez-Naranjo, “Two-hidden-layer feed-forward networks are universal approximators: a constructive approach,” Neural Networks, vol. 131, pp. 29–36, Nov. 2020, doi: 10.1016/j.neunet.2020.07.021.

[31] T. Deng, “Effect of the number of hidden layer neurons on the accuracy of the back propagation neural network,” in Proceedings of the 2023 4th International Conference on Medical Imaging, Sanitation and Biological Pharmacy (MISBP 2023), Osaka, Japan, Nov. 2023, pp. 462-468, doi: 10.54097/nbra6h45.

[32] C. Banerjee, T. Mukherjee, and E. Pasiliao, “An empirical study on generalizations of the ReLU activation function,” in Proceedings of the 2019 ACM Southeast Conference (ACMSE 2019), Association for Computing Machinery, New York, USA, Apr. 2019, pp. 164–167, doi: 10.1145/3299815.3314450.

[33] G. Luo, X. Wang, W. Zhao, , S. Tao, and Z. Tang, ”ReLU neural networks and their training,” Mathematics, vol. 14 no. 1, pp. 39/1-39/12, Jan 2026, doi: 10.3390/math14010039.

[34] S. Mehra, G. Raut, R. D. Purkayastha, S. K. Vishvakarma, and A. Biasizzo, “ An empirical evaluation of enhanced performance softmax function in deep learning,”, IEEE Access, vol. 11, pp. 34912-34924, Jan. 2023, doi: 10.1109/ACCESS.2023.3265327.

[35] M. Franke and J. Degen, “The softmax function: properties, motivation, and interpretation,” PsyArXiv, 28 Sept. 2023, doi:10.31234/osf.io/vsw47.

Double Deep Q-Network-Based Smart Channel Hopping for Reactive Jamming Mitigation in IoT Networks: A Reliability and Energy-Aware Approach

Enos Letsoalo[1], Topside Mathonsi[2], Tshimangadzo Tshilongamulenzhe[3], Daniel du Duplesis[4]
Department of Information Technology, Tshwane University of Technology, Pretoria, South Africa[1, 2, 3]
Department of Computer Science, University of Johannesburg, Johannesburg, South Africa[4]

***Abstract*—Reactive jamming attacks are prevalent in low power Internet of Things networks and are severely affecting the performance of IoT nodes. Existing anti-jamming studies focus primarily on communication reliability. Limited attention has been given to the joint optimization of packet delivery ratio (PDR), energy consumption, and network lifetime in mobile IoT networks under reactive jamming attacks. This study proposes a double deep Q-network (DDQN)-based smart channel hopping model (SCHM) that improves communication reliability while reducing energy expenditure and extending network lifetime. We use a Markov Decision Process (MDP) for modelling the interaction between the jammer and the legitimate IoT node. We develop a DDQN-based SCHM algorithm using the Python programming language. We implement the DDQN-based SCHM approach in the NS-3.41 environment. We evaluated the effectiveness of DDQN-based SCHM against Deep-Q-Network (DQN)-based hopping, non-hopping, and random hopping approaches. The results showed that our proposed DDQN-based SCHM approach achieved an average PDR of 99%, experienced the lowest energy consumption, and prolonged the network lifetime of IoT nodes. The proposed DDQN-based SCHM approach improves communication reliability and energy efficiency of IoT nodes.**

***Keywords*—DDQN; SCHM; MDP; reactive jamming attacks; PDR; energy consumption; network lifetime**

I. Introduction

Various application domains such as smart homes, smart cities, and environmental monitoring are beneficiaries of IoT network systems and sensor systems. IEEE 802.15.4 specifically benefits from IoT network systems [1], [2], [3]. For example, IEEE 802.15.4 enables users with wearable devices and ingestible sensors to be monitored remotely [4]. This helps healthcare professionals remotely monitor patients with chronic diseases for better disease management. IEEE 802.15.4 IoT networks are vulnerable to various attacks, specifically reactive jamming attacks that affect their performance severely.

IoT networks are more susceptible to various cyber-attacks because they inherit common characteristics of wireless networks, namely: open access and shared access. Exposure to several malicious attacks originates from factors such as IoT nodes having minimal or no security features embedded in them [5].

This is because IoT nodes have energy constraints imposed on them and have limited computational capabilities. Therefore, the fact that IoT nodes utilize a shared transmission medium is also a concern [6].

During a reactive jamming attack, an attacker listens to activities in the channels and transmits high signal power to the active channel with the intention of disrupting communication [7]. A disrupted IoT node will try to retransmit the packets because of a lack of acknowledgment from the receiver. Rapid retransmission of packets can lead to a quick battery depletion of IoT nodes [8], [9]. This will further lead to a short lifetime of IoT nodes in the network. Furthermore, prolonged transmission delays and low PDR may be incurred. In IEEE 802.15.4 IoT networks, applications such as remote patient monitoring (RPM) with wearables or ingestible sensors will be left unmonitored due to jamming attacks, leading to poor chronic disease management [4].

Wireless sensor networks (WSNs) are applications of IoT networks and are susceptible to reactive jamming attacks. Sensor nodes are deployed in unattended areas and can be easily compromised. Reactive jamming attacks are frequently employed by attackers to disrupt normal operations of WSNs. The disruption of communication through jamming often leaves areas unmonitored in WSNs. The study by [10] suggested that jamming of communication channels can be mitigated by raising the transmission power of legitimate nodes. However, increasing power levels may lead to quick battery depletion of IoT nodes, leaving systems unmonitored [11].

This research proposes a DDQN-based SCHM approach, which is applied to mitigating reactive jamming attacks in IoT networks. This study uses the MDP for the formulation of jammer and legitimate node interaction. DDQN-based SCHM server, which runs independently, is used for training the learning agent. This research combines IoT networks, specifically 802.15.4, PDR, energy consumption, channel hopping, and reinforcement learning in one NS-3.41 framework. The performance metrics of our proposed system are: PDR, energy consumption, network lifetime, and packet loss. The NS-3.41 testbed that was used includes nine IoT nodes, the access point as the sink node, and one reactive jammer. The effectiveness of the proposed SCHM approach was validated against the DQN-based hopping approach, non-hopping approach, and random hopping approach. The obtained simulation results demonstrated that the proposed DDQN-based SCHM has increased PDR by an average of 6%, reduced energy consumption by IoT nodes by an average of 77.5%, prolonged

network lifetime of IoT nodes by an average of 27.3%, and reduced packet loss by an average of 88.5% compared to benchmark approaches.

The rest of the study is organized as follows: Section II presents analysis of related existing studies on reactive jamming mitigation, Section III presents the system model and problem formulation using MDP, Section IV presents the proposed DDQN-based SCHM approach, Section V presents the simulation setting and results, and Section VI presents the conclusion and future work.

II. Related Work

Recent advances in wireless communication security have focused on mitigating jamming attacks using intelligent channel selection strategies [12], [13], [14]. Traditional anti-jamming techniques, such as frequency hopping and spread spectrum, rely on predefined patterns and often fail in dynamic or adversarial environments [11], [15].

To address these limitations, reinforcement learning (RL) has been widely adopted for adaptive anti-jamming communication. Deep Q-Networks (DQN) have been used to enable agents to learn optimal channel selection policies without prior knowledge of jammer behavior [16]. For example, deep reinforcement learning-based hopping strategies have demonstrated improved robustness against stochastic and wideband jamming attacks.

However, standard DQN suffers from overestimation bias, which can degrade performance in highly dynamic environments [17]. To overcome this limitation, DDQN has been proposed, showing improved stability and convergence in anti-jamming scenarios [18]. These approaches enable more reliable decision-making under uncertain channel conditions.

In parallel, research in cognitive IoT networks has emphasized the importance of jointly optimizing communication reliability and energy efficiency. Reinforcement learning has been applied to balance throughput maximization with energy consumption minimization, particularly under adversarial interference such as jamming attacks [19].

Despite these advances, most existing studies rely on abstract simulation environments and do not fully capture realistic network behavior. In contrast, this work implements a DDQN-based smart channel hopping strategy within an NS-3 simulated IoT network under reactive jamming conditions. The proposed approach evaluates both PDR and energy consumption, providing a comprehensive assessment of communication performance and network sustainability under adversarial conditions.

The main contributions of this study are as follows:

- A realistic NS-3-based IoT simulation environment incorporating mobile sensor nodes, a fixed access point, and a reactive jammer operating in an IEEE 802.15.4 network.
- A DDQN-based smart channel hopping mechanism capable of selecting communication channels dynamically in response to jamming activity.
- A comprehensive performance evaluation framework that jointly measures packet delivery ratio, packet loss, energy consumption, and network lifetime.
- A comparative analysis of four communication strategies: non-hopping communication, random channel hopping, DQN-based channel hopping, and DDQN-based SCHM approach.
- An investigation of the trade-off between communication reliability and energy efficiency under reactive jamming conditions.

III. System Model and Problem Formulation

A. System Model

The simulation environment is implemented in NS-3.41, modeling an IoT network consisting of multiple mobile sensor nodes communicating with a fixed access point (AP). The IoT network consists of a set of transmission channels (C_1, C_2, …, C_n). A reactive jammer is introduced, which transmits interfering signals upon detecting legitimate communication. Each IoT node generates UDP traffic toward the AP using an IPv6 over 6LoWPAN stack. Node mobility is modelled using a random walk mobility model within a bounded area, while the jammer also follows a dynamic movement pattern. Fig. 1 shows the proposed system model for the IoT network.

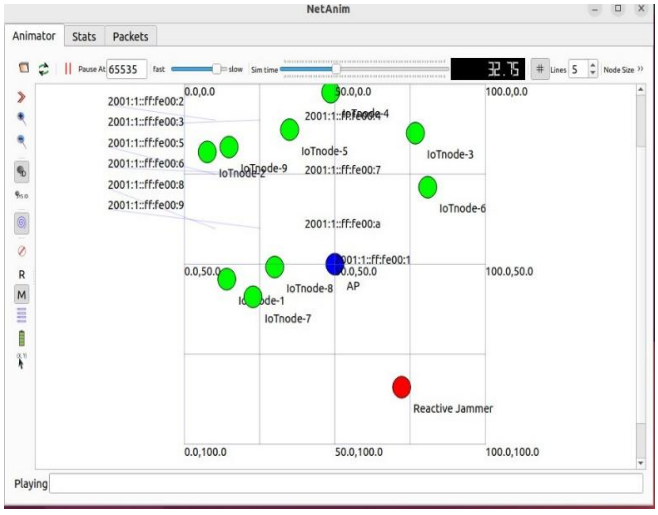


Fig. 1. System model for IoT network.

To mitigate reactive jamming effects, a DDQN-based SCHM agent is employed to perform smart channel hopping across available IEEE 802.15.4 channels. The agent observes the network state, including PDR, signal quality indicators, and energy consumption, and selects channel actions to maximize communication reliability and energy efficiency.

B. Jammer Model

The reactive jammer positions itself between the legitimate IoT node, sender (S), and the AP, receiver (R), and interrupts the communication by injecting a high-power signal. The actual transmitted signal by legitimate IoT nodes is affected by the normal noise known as Additive White Gaussian Noise (AWGN). Therefore, the optimal received signal at the

receiver's end without the presence of the jammer is given by Eq. (1):

$$R(t) = s(t) + g(t) \quad (1)$$

where, R(t) denotes the signal that is received, s(t) is the actual signal transmitted by the sender, and g(t) is the AWGN that can be between the path of the sender and the receiver.

The jammer can typically interrupt the communication by injecting the signal $s_j(t)$ into the active transmission channel. Therefore, the actual received signal at the receiver's end with the jammer in the vicinity is given by Eq. (2):

$$R(t) = s(t) + s_j(t) + g(t) + g_1(t) \quad (2)$$

where, $s_j(t)$ represents the actual jammer signal, and $g_1(t)$ is the AWGN between the jammer and the receiver. This research study uses Received Signal Strength (RSS) and PDR for detecting reactive jamming attacks. These parameters are estimated by most wireless communication systems that are equipped with transmitters and receivers.

C. Problem Formulation

Reinforcement learning requires the formulation of a Markov Decision Process (MDP). Therefore, the proposed SCHM approach adopts the MDP framework for the formulation of the optimization problem of the system. The MDP consists of the state space, action space, and the reward [20]. Typically, the agent observes the state space, which is the environment, and selects an action and eventually receives a reward accordingly.

1) State space: The IoT node is assumed to know nothing about the environment that it is interacting with, i.e., the status of the channel, the strategy of the jammer, the number of frames in the transmission queue, and the capacity of energy storage. The legitimate IoT node can only observe the chosen channel for communication. The state space is defined by Eq. (3).

$$S = \{(c, j, q, e): c \in (0,1), j \in (0,1), q \in (0, \dots, Q), e \in (0, \dots, E)\} \quad (3)$$

where, communication channel, $c = 0$, $c = 1$, denotes that the communication channel is idle and occupied, respectively. Jammer, $j = 0$ denotes that there is no attack on the communication channel and $j = 1$ that a jammer is attacking a communication channel. q denotes the packets in the transmission queue and e denotes the energy storage of the IoT node.

2) Action space: The IoT node can execute any of the following actions, namely, remain idle, actively transmit packets, or hop into another free radio channel and transmit packets when there is a jamming attack in the currently selected channel. The state space is defined by Eq. (4).

$$A = (a: a \in \{1,2,3\}) \quad (4)$$

where, 1 denotes that the IoT node remains idle, 2 denotes that the IoT node transmits packets, and 3 denotes that the IoT node hops into the unjammed communication channel.

3) Reward function: The immediate reward for the system is defined in this section. It is described as the number of packets that are sent to the receiver and positively acknowledged. The actions that the IoT node takes must reduce packet loss, which will increase PDR. Furthermore, the actions must reduce energy consumption, which will prolong the network lifetime of the IoT network. The reward function is defined in Eq. (5).

$$r(s, a) = \{pkt_{sen_{ack}}, hop_{suc}, avoid_{jam_{ch}}\} \quad (5)$$

where, $pkt_{sen_{ack}}$ denotes the packets that were sent and acknowledged successfully, hop_{suc} denotes that the learning agent hops successfully into an unjammed channel, and $avoid_{jam_{ch}}$ denotes that the learning agent avoids transmitting into the jammed channel.

IV. Proposed DDQN-Based SCHM Approach

A. Design of DDQN-Based SCHM

In this section, this research presents the development of a DQN algorithm for approximating the solution to the MDP problem formulated in Section III. The Q-learning algorithm, which is based on creating a reference table for selecting optimal actions and eventually storing state-action values in a table, cannot handle the jamming mitigation problem formulated in this research because the state space size is too large [21]. Consequently, this research study employs a DQN-based SCHM to fast-track the learning process and attain the best defence policy without being aware of the channel status and jamming strategy of the jammer.

This research adopts the Double-DQN method for approximating the Q-function. The reason is that DQN alone has a problem of performance instability due to a non-linear function [22]. DDQN utilizes two sets of neural networks, namely a policy network and a target network. DQN utilizes a target network to select and evaluate the next action. DDQN uses a policy network to select the next action and a target network to evaluate the action taken. Fig. 2 illustrates the proposed DDQN-based smart channel hopping strategy.

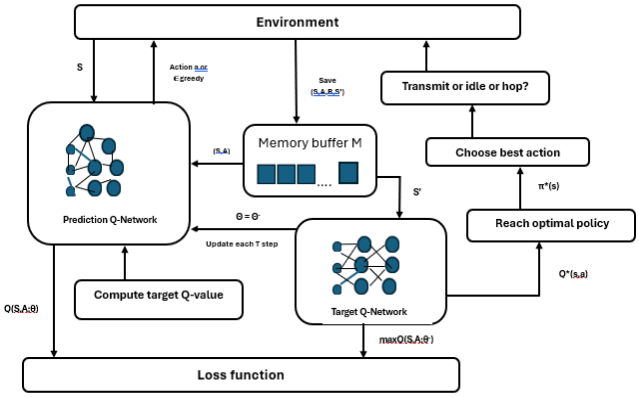


Fig. 2. Proposed DDQN-based SCHM approach.

To demonstrate the process, this research designs the DDQN-based smart channel hopping strategy shown in Algorithm 1. There are basically three significant elements of

the DDQN framework, namely the prediction Q-network $Q(s_t, a_t, \theta)$, the target Q-network $Q(s_t, a_t, \theta^-)$, and the memory buffer B.

The prediction network $Q(s_t, a_t, \theta)$ is used to approximate the Q-value; then the target Q-network $Q(s_t, a_t, \theta^-)$ and the memory buffer B are used to ensure that the DDQN is stable and that the target Q-value does not oscillate. The DDQN algorithm's advantage over the Q-learning algorithm is the ability to record the transition (s_t, a_t, r_t, s_{t+1}), i.e., the current state, the action, the immediate reward, and the next state observed by the legitimate node, in a memory buffer B and then train the Q-network every epoch time slot. During Q-network training, buffered transitions are sampled randomly from a memory buffer and are used to compute the corresponding target Q-value. Table I gives a summary of the symbols and their description.

Algorithm 1: DDQN-Based SCHM Algorithm

1. Initialization of prediction Q-network **Q** with random weight θ, target Q-network $\mathbf{Q}^-$ with random weight θ^- and the replay memory B with capacity M
2. for episode = 1,2, 3,..,E **do**
   Select two channels randomly from set C to initialize the state S, where N channels are used for communication and the other channel for hopping
3. for timeslot t = 1 to T **do**
   Choose a random action a_t with probability ∈, otherwise choose $a_t = \text{argmax}Q(s_t,a_t,\theta)$
4. Execute action a_t and receive the immediate reward r_t, then obtain next state s_{t+1}
5. Save transition (s_t, a_t, r_t, s_{t+1}) in replay memory B
6. Select random samples (s_i, a_i, r_i, s_{i+1}) from replay memory B
7. Calculate target Q-value using the Q-value function below
   $Y_j = r_j + \gamma Q(s_{j+1}, \text{argmax}_a Q(s_{j+1},a;\theta); \theta^-))$
8. The weights are then optimized by applying gradient descent using parameter θ to minimize the loss using the equation below
   $(y_j - Q(s_j,a_j; \theta))^2$
9. Repeat steps 6 to 8.
10. Replace $\mathbf{Q}^-$ with **Q** after a fixed number of steps
11. end for
12. end for

TABLE I. SYMBOLS AND THEIR DESCRIPTION

Symbol	Description
Q*(s, a)	Q-function given a state and action
Q_θ	Prediction network
Q_{θ^-}	Target network
t	timeslot
S	State space
A	Action space
s_t	State of the environment at timeslot t
a_t	Action taken at time slot t
r_t	Learning agent reward at time slot t
s_{t+1}	State of the environment at timeslot t+1
γ	Discount factor
∈	Epsilon

V. SIMULATION RESULTS

A. Simulation Setting

This research work implements an NS-3.41-based IoT simulation for evaluating reactive jamming and reinforcement-learning-based channel hopping. The simulation models mobile IoT nodes, a fixed access point, a reactive jammer, packet delivery ratio, packet loss, per-node energy consumption, and network lifetime. The reinforcement learning component is implemented separately in Python using a DDQN agent that selects channel-hopping actions based on network feedback from the NS-3.41 simulation. The plotting component is also separated into an independent Python script to compare non-hopping, random hopping, DQN-based hopping, and DDQN-based SCHM hopping scenarios. The contribution of this research is the integration of reactive jamming, channel hopping, DDQN decision-making, and performance evaluation into one experimental framework for studying communication reliability and energy efficiency in IoT networks under jamming attacks. RL parameters for simulation are shown in Table II.

TABLE II. RL PARAMETERS AND VALUES

Parameter	Value
Learning rate	1e-3
Gamma	0.98
buffer_size	20000
batch_size	64
target_update	100
Epsilon_start	1.0
Epsilon_min	0.05
Epsilon_decay	0.995

The performance metrics that were used in this research are energy consumption, PDR, network lifetime, and packet loss.

- Energy consumption – The amount of energy that the IoT node uses for transmitting or receiving packets. IoT nodes consume more energy during jamming attacks because of the need for retransmission when there is a lack of acknowledgment.
- PDR – Measures the number of packets that are successfully received at the destination in relation to the total packets sent at the source. Packet loss due to jamming attacks can significantly lead to poor PDR.
- Network lifetime – Refers to the duration of time in the LR-WPAN where all the nodes are active in the network during simulation and still have energy to participate in transmission of packets. This essentially measures the energy consumption of IoT nodes during the simulation.
- Packet loss – Occurs when data packets fail to reach the destination in the network. Interference caused by

reactive jamming attacks can result in severe packet loss in IoT networks.

B. Simulation Scenarios

Simulations are conducted under four scenarios, namely the non-hopping approach, which is a baseline, random hopping approach, DQN-based hopping approach, and DDQN-based SCHM approach, which is the proposed mitigation strategy.

- Non-channel hopping (baseline) where the IoT nodes use a single communication channel during transmission.
- Random channel hopping where the IoT nodes periodically switch the communication channels during transmission.
- DQN-based channel hopping, which uses a policy-based approach to send actions to the IoT nodes for switching to a new communication channel.
- DDQN-based SCHM approach where the learning agent receives network feedback from the network and selects actions and sends the signal to the IoT nodes for switching into the new communication channel.

The results are compared to assess the effectiveness of the proposed approach in improving network performance under reactive jamming conditions.

C. Results

When assessing the performance of four hopping strategies, the simulation period was configured in the NS-3.41 environment such that it runs for 220 seconds. Table III shows the simulation parameters and their values that were set on NS-3.41 environment during the simulation period.

TABLE III. SIMULATION PARAMETERS AND VALUES

Parameter	Value
BurstPackets	600
PacketSize	80 bytes
simulationTime	220 seconds
jamStartTime	20 seconds
jamEndTime	180 seconds
InitialEnergy	5.0 Joules
TransmitPowerW	60 Milliwatts
ReceivePowerW	50 Milliwatts
IdlePowerW	1 Milliwatts
FrequencyBand	2.4 GHz
NumOfChannels	16 (11 – 26)
JammerDetectionMetrics	RSS and PDR

To ensure methodological precision and improve the statistical reliability of the findings, each experimental scenario is repeated 20 times independently using the same simulation parameters. The results from these runs are aggregated by computing the mean PDR, while variation across runs is examined using appropriate dispersion measures such as standard deviation. Results are reported as mean ± standard deviation over 20 independent simulation runs, as shown in statistical summary Table IV. The mean PDR is recorded in percentage, mean packet loss is recorded in number of packets lost, mean energy is recorded as total energy consumed, and mean network lifetime is recorded in seconds relative to total simulation time.

TABLE IV. STATISTICAL SUMMARY

Scheme	PDR (%)	Packet Loss (number)	Energy (J)	Network Lifetime (s)
Non-hopping	89 ± 0.05	212 ± 1.89	8.5 ± 0	132 ± 0.72
Random hopping	93 ±1.28	134 ± 14.97	6.8 ± 1.39	172 ± 25.02
DQN	97 ± 1.25	5.4 ± 1.47	Mean ± SD	176 ± 26.12
DDQN	99 ± 0.57	9 ± 4.93	1.5 ± 0.47	220 ± 0

1) PDR: During the performance evaluation, the proposed SCHM approach, DQN hopping approach, random hopping approach, and non-hopping approach underwent mean PDR of 99%, 97%, 93%, and 89%, respectively, as displayed in Fig. 3. The proposed DDQN-based SCHM approach outperforms the DQN hopping approach by 2%. In addition, the proposed DDQN-based SCHM approach outperforms the random hopping approach by 6% PDR at the end of the simulation. Furthermore, the proposed DDQN-based SCHM approach outperforms the non-hopping approach by 10% PDR at the end of the simulation. The proposed approach significantly improves communication reliability as compared to non-hopping and random hopping approaches.

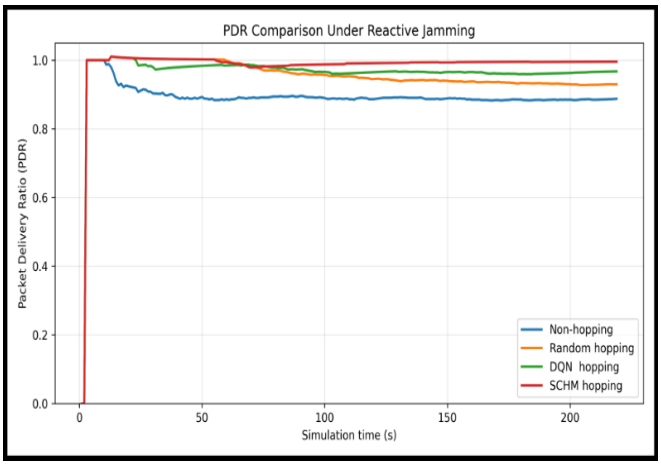


Fig. 3. PDR comparison under reactive jamming attack.

2) Energy consumption per node: All nodes, including the AP, are initially allocated the same energy budget of 5 Joules. During the performance evaluation, the average energy consumption per node for the proposed DDQN-based SCHM approach, DQN hopping approach, random hopping approach, and non-hopping approach is 1.5 Joules, 5.4 Joules, 6.8 Joules, and 8.5 Joules, respectively, as displayed in Fig. 4. Compared to the DQN hopping approach, the proposed DDQN-based

SCHM approach reduced energy consumption by 72.2%. Compared to the random hopping approach, the proposed SCHM approach reduced energy consumption by 77.9%. Compared to the non-hopping approach, the proposed DDQN-based SCHM approach reduced energy consumption by 82.3%. The proposed DDQN-based SCHM approach significantly improved the energy consumption of IoT nodes by the end of the simulation period. Random hopping and non-hopping strategies did not utilize energy efficiently, as shown in Fig. 4. All the nodes depleted the entire 5 joules of energy allocated to each of them at the end of the simulation.

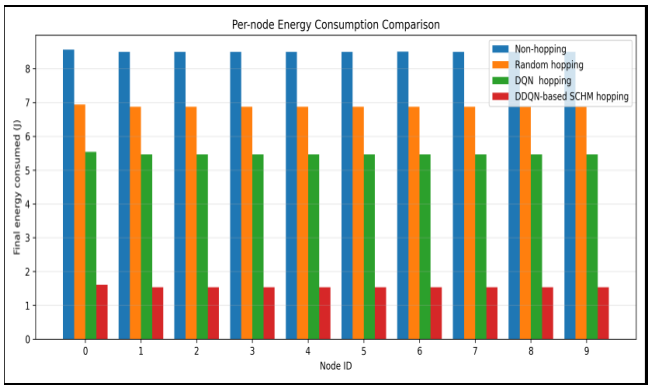


Fig. 4. Energy consumption comparison under reactive jamming attack.

3) Network lifetime: During the performance evaluation, the DQN hopping approach, random hopping approach, and non-hopping approach recorded the mean network lifetime of 176 seconds, 172 seconds, and 132 seconds of the simulation time, respectively. The proposed SCHM approach did not record any death of IoT nodes; all IoT nodes are alive by end of simulation period, hence the mean network lifetime recorded is 220 seconds. Compared to the DQN hopping approach, the proposed SCHM approach extended the network lifetime by 20% during the simulation period. Compared to the random hopping approach, the proposed SCHM approach extended the network lifetime by 21.8% during the simulation period. Compared to the non-hopping approach, the proposed SCHM approach extended the network lifetime by 40% during the simulation period. The proposed SCHM approach experienced the longest span in terms of network lifetime, as displayed in Fig. 5. DQN hopping, random hopping, and non-hopping strategies experienced short to medium durations of network lifetime. IoT nodes' energy is depleted before the simulation time expires, as shown in Fig. 5. The proposed SCHM approach extends network lifetime compared to DQN hopping, random hopping, and non-hopping approaches.

4) Packet loss: During the performance evaluation, the mean number of packet losses for the proposed DDQN-based SCHM approach, DQN hopping approach, random hopping approach, and non-hopping approach are 9 packets, 38 packets, 137 packets, and 212 packets, respectively, as demonstrated in Fig. 6. Compared to the DQN hopping approach, the proposed DDQN-based SCHM approach reduced packet loss by 76.3% during the simulation period. Compared to the random hopping approach, the proposed SCHM approach reduced packet loss by 93.4% during the simulation period. Compared to the non-hopping approach, the proposed DDQN-based SCHM approach reduced packet loss by 95.7% during the simulation period. The proposed DDQN-based SCHM approach experienced the lowest number of packet losses during the simulation, as shown in Fig. 6. The DQN hopping strategy experienced the lowest number of packet losses during the simulation. The random hopping strategy experienced a medium number of packet losses during the simulation. The non-hopping strategy experienced the highest number of packet losses, as displayed in Fig. 6.

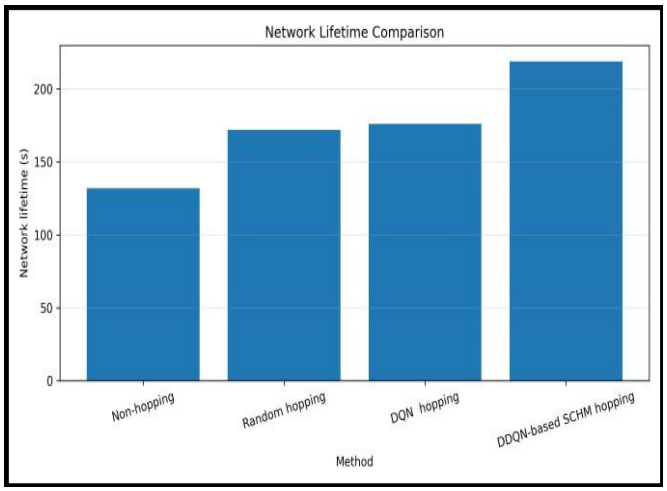


Fig. 5. Network lifetime comparison under reactive jamming attack.

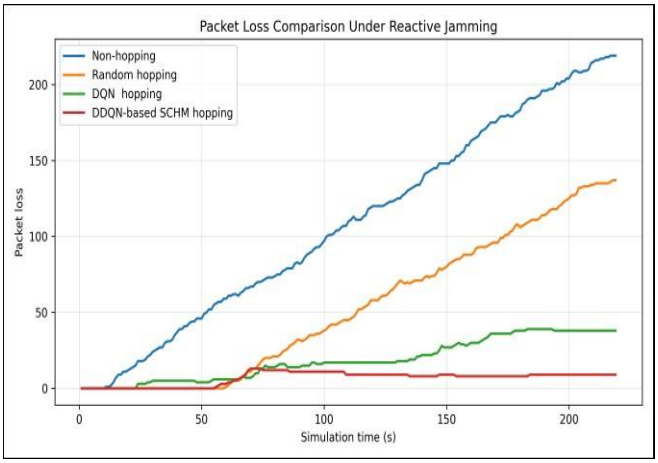


Fig. 6. Packet loss comparison under reactive jamming attack.

D. Discussion

The results demonstrate that the DDQN-based SCHM approach significantly improves communication reliability under reactive jamming conditions. Compared to the baseline scenario without channel hopping, the proposed approach achieves a higher PDR and reduces packet loss. This is shown in Fig. 3 and Fig. 6.

Furthermore, the DDQN-based SCHM agent exhibits improved energy efficiency by avoiding heavily jammed

channels, thereby reducing unnecessary retransmissions. This is shown in Fig. 4. This leads to a slower rate of energy depletion across IoT nodes and extends overall network lifetime. This is shown in Fig. 5.

When compared to random channel hopping, the DDQN-based SCHM approach shows more stable performance and faster adaptation to jammer behavior. These findings are consistent with existing literature, which highlights the advantages of deep reinforcement learning in dynamic and adversarial wireless environments.

This research study compares the proposed DDQN-based SCHM approach with the RL algorithm proposed by [16] and the DDQN algorithm proposed by [18]. The results of the study by [16] show that DRL-based hopping outperforms classical Q-learning and is robust even when jammer behaviour is unknown. The results of the study by [18] show that DDQN fixes overestimation problems, better stability and performance and DDQN performs better and more stable than DQN. The key validation factors are listed in Table V.

TABLE V. VALIDATION FACTORS

Metric	Paper [16]	Paper [18]	This research study
Throughput / PDR	Yes	Yes	Yes
Channel selection policy	RL-based	DDQN	DDQN
Jamming type	Probabilistic	Sweeping	Reactive
Performance improvement	DRL better than Q-learning	Fix overestimation	DDQN-based vs baseline
Limitations	Q-value overestimation	Did not evaluate energy or lifetime	Evaluates PDR, packet loss, energy, network lifetime and convergence

Overall, the proposed DDQN-based SCHM approach demonstrates the effectiveness of intelligent channel selection in enhancing both reliability and energy efficiency in IoT networks under jamming attacks.

VI. CONCLUSION

This research evaluated the performance of the proposed DDQN-based SCHM approach under reactive jamming attacks against DQN-based hopping, non-hopping, and random hopping approaches. The proposed DDQN-based SCHM approach was implemented using NS-3.41 with the IEEE 802.15.4 protocol, simulating an IoT network. The simulation results showed that the DDQN-based SCHM approach achieves better convergence in terms of reward, PDR, and loss. During the performance evaluation the following metrics were used namely: PDR, packet loss, energy consumption, and network lifetime. The simulation results demonstrated that the DDQN-based SCHM approach can improve PDR and reduce packet loss as compared to DQN-based hopping, non-hopping, and random hopping approaches. The simulation results showed that the proposed DDQN-based SCHM has improved communication reliability of the 802.25.4 IoT network. The results are consistent with similar studies that utilized DDQN.

Furthermore, the simulation results showed that the DDQN-based SCHM approach can significantly reduce energy consumption and prolong the network lifetime as compared to DQN-based hopping, non-hopping, and random hopping approaches. The simulation results showed that the proposed DDQN-based SCHM has addressed the shortcomings of the existing literature in similar studies. The existing literature has given little attention to energy consumption and network lifetime of IoT nodes and has focused on detection and communication reliability. As future work, the proposed DDQN-based SCHM approach can include performance improvements in QoS and channel access to reduce collisions. Future extension of the study will also evaluate the performance of the proposed approach in a scenario where multiple reactive jammers exist. In addition, future work can adopt the solution and evaluate it in multichannel jamming, UAV, and satellite navigation systems.

REFERENCES

[1] Fourati, S. Khssibi, T. Val, H. Idoudi, a. Van Den Bossche, and L.A. Saidane,“Comparative Study of IEEE 802.15. 4 and IEEE 802.15. 6 for WBAN-based”, CANet,November 2015

[2] Iova, F. Theoleyre, T. Watteyne, and T. Noel, “the Love-hate Relationship between IEEE 802.15. 4 and RPL”. IEEE Commun. Mag. 55(1), P.188–194, 2016.

[3] Ahmed, H. Rahman, and M.I. Hussain, “A Comparison of 802.11 Ah and 802.15. 4 for IoT”. ICT Express 2(3), P. 100–102, 2016.

[4] C. Fourati, and S. Said, “Remote Health Monitoring Systems Based on Bluetooth Low Energy (BLE) Communication Systems” ICOST 2020, LNCS 12157, Pp. 41–54, 2020.

[5] Zahra, Y.S. Bostanci, and M. Soyturk, “Real-Time Jamming Detection in Wireless IoT Networks”, IEEE Access, Volume 11, 2023, P. 70425 – 70442.

[6] Mukherjee, ‘‘Physical-layer Security in the Internet of Things: Sensing and Communication Confidentiality under Resource Constraints,’’ Proc. IEEE, Vol. 103, No. 10, Pp. 1747–1761, Oct. 2015.

[7] Abdullahi, Y. Baashar, H. Alhussian, a. Alwadain, N. Aziz., L.F. Capretz, and S.J. Abdulkadir, “Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: a Systematic Literature Review”, Electronics, Vol. 11, No. 2, P. 198, 2022.

[8] Chen, Y. Li, D. Xu, and L. Xiao, “DQN-based Power Control for IoT Transmission against Jamming”. IEEE 87th Vehicular Technology Conference (VTC Spring), 2018.

[9] Chkirbene, R. Hamila, and a. Erbad, “Secure Wireless Sensor Networks for Anti-Jamming Strategy Based on Game Theory”. 2023 International Wireless Communication and Mobile Computing (IWCMC), 2023.

[10] Vadlamani, B. Eksioglu, H. Medal, and a. Nandi, “Jamming Attacks on Wireless Networks: a Taxonomic Survey,” International Journal of Production Economics, Vol. 172, Pp. 76–94, 2016.

[11] Strasser, B. Danev, and S. C. Capkun, “Detection of Reactive Jamming in Sensor Networks,” ACM Transactions on Sensor Networks (TOSN), Vol. 7, No. 2, P. 16, 2010.

[12] Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. V. Poor, “Two-dimensional Antijamming Mobile Communication Based on Reinforcement Learning,” IEEE Transactions on Vehicular Technology, Vol. 67, Pp. 9499–9512, Oct. 2018.

[13] Liu, Y. Xu, L. Jia, Q. Wu, and a. Anpalagan, “Antijamming Communications Using Spectrum Waterfall: a Deep Reinforcement Learning Approach,” IEEE Communications Letters, Vol. 22, Pp. 998–1001, May 2018.

[14] Gao, Z. Qin, X. Jing, Q. Ni, and S. Jin, “Antiintelligent UAV Jamming Strategy via Deep Q-networks,” IEEE Transactions on Communications, Vol. 68, Pp. 569–581, Jan. 2020.

[15] Thamilarasu and R. Sridhar, “Game Theoretic Modeling of Jamming Attacks in Ad Hoc Networks,” Proceedings of 18th International

Conference on Computer Communications and Networks (ICCCN), Pp. 1–6, 2009.

[16] Qi, H. Zhang, X. Qi, and M. Peng, "Deep Reinforcement Learning Based Hopping Strategy for Wideband Anti-Jamming Wireless Communications". IEEE Transactions on Vehicular Technology, Vol. 73, No. 3, March 2024.

[17] Thrun and a. Schwartz, "Issues in Using Function Approximation for Reinforcement Learning," 10 1993.

[18] K. Nguyen, D.H. N. Nguyen, N.H. Tran, and D. Brunnenmeyer, "Deep Reinforcement Learning Approaches against Jammers with Unequal Sweeping Probability Attacks". EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, Volume 12, Issue 4, Pages 1 -9, 2025.

[19] Abdolkhani, N.A. Khalek, and W. Hamouda, "Deep Reinforcement Learning for EH-Enabled Cognitive-IoT under Jamming Attacks". IEEE Internet of Things Journal, Vol. 11, No. 24, Pages 40800 – 40813, December 2024.

[20] Slimeni, F., B. Scheers, Z. Chtourou, and V. Le Nir, "Jamming Mitigation in Cognitive Radio Networks Using a Modified Q-learning Algorithm", in Proc. IEEE Int. Conf. Mil. Commun. Inf. Syst., 2015, Pp. 1–7.

[21] Fährmann, D., Jorek, N., Damer, N., Kirchbuchner, F., & Kuijper, F. (2022). Double Deep Q-Learning With Prioritized Experience Replay for Anomaly Detection in Smart Environments. IEEE Access. Volume 10.

[22] Huynh, N., Nguyen, D.N., Hoang, D.T., and Dutkiewicz, E. (2021). Defeating Reactive Jammers with Deep Dueling-based Deception Mechanism. ICC 2021 - IEEE International Conference on Communications.

Comparative Evaluation of Synthetic Data Generation Methods for Binary Classification Problems with Class Imbalance

Jhonatan Esquivel, Christian Humpiri, Jose Vega, Nemias Saboya
Escuela Profesional de Ingeniería de Sistemas-Facultad de Ingeniería y Arquitectura,
Universidad Peruana Unión, Lima, Perú

***Abstract*—Class imbalance is a critical challenge in the classification of tabular data, since it affects the diagnostic capacity of models in domains such as health and finance. This research compares four synthetic data generation paradigms: traditional interpolation (SMOTE-NC), deep generative models (CTGAN and TVAE), and a hybrid scheme (SMOTE-NC-CTGAN) in which SMOTE-NC only densifies the minority data used to train the generator. Five UC Irvine datasets with imbalance ratios between 2.55:1 and 29.99:1 were used under a dual evaluation that combines multidimensional statistical fidelity (Wasserstein distance, correlation and categorical differences, and diversity) and predictive utility (AUC-PR, Recall, Precision, F1-Score, Specificity and G-mean), with a cost-sensitive baseline as reference. Each experiment was repeated in 10 runs that regenerate the synthetic data, and the differences were tested with paired tests, Holm correction, and effect sizes. SMOTE-NC obtains the highest fidelity in all datasets, while the CTGAN-based methods generate the most dispersed records under extreme scarcity. Synthesis does not improve the global ordering of predictions: of 40 comparisons, AUC-PR decreases significantly in 19 and increases only in 2, both in the most imbalanced dataset with XGBoost (HYBRID: 0.5252 versus 0.4872), an effect not observed with CatBoost. In contrast, Recall increases significantly in 27 of the 40 comparisons (e.g., from 0.1617 to 0.5835 with CTGAN in DS1), at the cost of Precision, and a class-weighted baseline produces a similar displacement. Data synthesis mainly shifts the decision boundary towards the minority class, which is favorable where false negatives dominate the cost.**

***Keywords*—Class imbalance; synthetic data; SMOTE-NC; CTGAN; hybrid models; machine learning; tabular classification**

I. Introduction

The imbalance of classes in datasets represents a very common challenge in machine learning tasks, as well as in the real world, where events of interest occur with low frequency. Consequently, classification algorithms often produce bias by favoring the majority class based on the level of imbalance. This problem has been addressed in various domains such as computer security, medical diagnosis, and detection of financial fraud, where incorrect detection can have critical consequences [1], [2], [3].

To address this problem, various approaches have been proposed, among which the oversampling of the minority class has been widely used. The methods derived from the Synthetic Minority Oversampling Technique (SMOTE) are effective and less computationally expensive with respect to other models. Likewise, they highlight how versatile they can be [2],[4]. However, these approaches have inherent limitations, since the generation of examples by means of linear interpolation can give rise to less diverse examples and does not adequately capture the complex relationships between variables.

Therefore, in recent years, different deep generative models have been created as a promising alternative for the generation of more realistic synthetic data [5]. In particular, Generative Adversarial Networks (GANs) have demonstrated a remarkable ability to learn complex data distributions and generate new samples that preserve the statistical characteristics of the original dataset [6]. These models have been applied in various domains and with different variations according to the context, such as CTGAN [7], CopulaGAN [8], CGAN [6], or AC-GAN [6].

Likewise, there are Variational Autoencoders (VAE), which seek to minimize the difference between the input and the reconstructed output, thus learning a compact representation of the data [8]. It presents multiple variants, and among them the most applied in the TVAE is a variation aligned to tabular data, which captures that, despite being highly sensitive to its hyperparameters, is a robust model whose performance varies according to the characteristics of the dataset [9]. However, the use of synthetic data introduces new challenges related to the quality and fidelity of the data generated, since some studies have indicated that the predictive capacity of models trained with synthetic data is not sufficient to evaluate their usefulness, as the statistical distributions and relationships between variables present in the original dataset must also be analyzed [10], [11].

More recently, some works have proposed hybrid approaches that combine traditional oversampling techniques with generative models. These strategies seek to take advantage of both approaches. On the one hand, the initial balancing facilitates the learning of the generative model. In contrast, the generative process allows the creation of more diverse and representative samples of the original distribution of the data [12], [13].

In this context, the present study aims to evaluate comparatively different synthetic data generation techniques for imbalanced binary classification problems. For this, traditional oversampling methods and generative models designed for tabular data are analyzed, including SMOTE-NC, CTGAN, and TVAE, as well as a hybrid SMOTE-NC-CTGAN approach. The

experiments are performed on multiple real datasets with unique characteristics, which allows evaluation of the behavior of these methods under different levels of imbalance.

The contributions of this work are: 1) a controlled comparison of interpolation, adversarial, variational, and hybrid generation under a common pipeline, with the same real records and classifiers; 2) a hybrid SMOTE-NC-CTGAN scheme for mixed-type data in which SMOTE-NC only densifies the minority data used to train the generator; 3) a dual evaluation of multidimensional fidelity and predictive utility, including a cost-sensitive baseline; and 4) a statistical validation over repeated runs that regenerate the synthetic data.

II. Literature Review

The problem of class imbalance has been widely studied since it is very relevant to machine learning models because this unequal distribution can produce bias or unjustified benefit towards the majority class, reducing its veracity. In the same way, recent literature has responded to this problem with various forms of oversampling, such as traditional techniques, generative models, variants, or hybrid approaches. This framework is responsible for precisely this: being able to compare and evaluate their fidelity against binary classification datasets with different levels of imbalance.

A. Class Imbalance in Tabular Data Sets

The literature agrees that class imbalance negatively affects models since many learn with a certain bias; consequently, they end up favoring the majority class. The study [2] summarizes that this problem is particularly critical in health, where the minority class usually corresponds to the most important clinical event. In a complementary way, the review [4] shows that both variants of SMOTE and approaches based on GAN have arisen precisely to correct this limitation in imbalanced classification tasks. Along the same lines, the work [3] points out that, in Android malware detection, the imbalance reduces the ability to generalize to the minority class. At the same time, [14] shows that even deep networks can remain biased if the training objective does not adequately control the unequal distribution between classes.

Therefore, generative models become relevant, and this is demonstrated in the study [15], where it is explicitly stated that imbalanced data sets hinder the development of robust models for predictions of obesity levels and that synthetic generation can efficiently support when samples or certain key variables are missing. However, the study [11] mentions that these methods must be analyzed quite carefully, because they can generate differences in distributions, correlations, or even privacy-related risks, especially when it is directly assumed that synthetic data will always be useful. In addition, in another context, the study [16] explains that in imbalanced regression problems, it is not enough to only increase the number of rare values, but it is also important to preserve the relationship between the variables and the continuous objectives. This shows that generating new data alone does not guarantee good results if the existing logic between the original data is not maintained.

B. Traditional Oversampling Techniques

Within traditional techniques, SMOTE continues to be one of the most used methods to treat the imbalance problem. Its operation is based on creating new samples of the minority class through interpolation between close neighbors, simply avoiding copying existing instances. The work [17], applied to the area of phishing detection, demonstrates that the combination of SMOTE-NC together with XGBoost achieved 98.0% accuracy and 98.5% recall, surpassing the model trained without re-sampling. This shows that data balancing can significantly improve the ability to detect the minority class.

Similarly, the study [18] concludes that the combination of SVM + SMOTE-NC presents better results compared to SVM without balancing techniques, especially in data sets containing nominal and continuous variables. In these cases, improvements in both accuracy and F1-score were observed. On the other hand, some applied works also support the usefulness of these techniques. In the case of credit card fraud detection, [1] reports a fairly extreme scenario where fraud only accounts for 0.172% of transactions made. In that study, the combination of Random Oversampling with XGBoost obtained the best performance, reaching an F1-score of 92.43%, accuracy of 97.383%, and recall of 88.418%. Likewise, the study [19] shows that after applying SMOTE to correct an imbalance ratio of 60:40 in zygosity prediction, XGBoost was again positioned as the model with the best results, achieving 85.57% accuracy and 86.72% F1-score.

In general, the studies analyzed suggest that traditional oversampling techniques still maintain a fairly competitive performance, especially when combined with robust classifiers. In addition, they allow for improving the detection of the minority class without increasing the complexity of the pipeline too much. However, studies [4] and [16] also point out that these methods can generate overly generalized or unrealistic samples when the feature space has complex structures. Due to this, their ability to represent heterogeneous tabular distributions may be limited in some scenarios.

C. Generative Models for Tabular Data

The advancement of generative models has opened the possibility of going beyond SMOTE's own local interpolation. The review [6] highlights that GANs have gained relevance as tools to alleviate imbalance in tabular datasets when modeling complex distributions. In this line, CTGAN has become one of the most used models because it was designed to work with mixed and imbalanced tabular data. The study [10] evaluated CTGAN using the Cleveland heart disease dataset and found that the model manages to reproduce the general distribution and some important correlations, especially in variables such as age, thalach, and oldpeak. However, there are still differences in certain categorical attributes. In addition, [10] mentions that the predictive usefulness of synthetic data is moderate, since although the data generated captures relevant patterns, they fail to replicate the original labels fully.

Similarly, the work [20] compared CTGAN and TVAE in tabular datasets of different sizes and concluded that both models can retain statistical properties. However, their performance depends quite a bit on the size of the data and the configuration used during training. On the other hand, [21] presents an improved conditional GAN architecture for cardiovascular data and reports a mean KS of 0.3900, close to TVAE (0.3858) and higher than CTGAN (0.4803). This shows that the behavior of generative models can vary depending on the characteristic evaluated. In fact, the study points out that some variables are better represented by TVAE and others by the proposed model, so there is no single dominant model. This idea coincides with [11], where TVAE is described as a fairly robust and stable method even in complex datasets with many categorical variables. However, it is also mentioned that it may present greater privacy risks, as some synthetic data tends to look too much like actual records.

D. Hybrid Approaches to Synthetic Generation

On the other hand, a particularly relevant line for this research is that of hybrid methods, which try to combine the simplicity of classic oversampling with the ability of generative models to learn more complex distributions. Article [12] is central at this point, since it proposes SMOTified-GAN, where the samples generated by SMOTE replace random noise as input to the GAN. The work [12] reports that this approach improved the F1-score by up to 9% with respect to the next-best method and that it surpassed the other algorithms in 10 out of 11 datasets, showing especially clear advantages in sets with very small minority classes.

Subsequent evidence goes in the same direction. The study [22] focused on financial fraud and compared SMOTE, GAN, SMOTified-GAN, SMOTE+GAN, and GANified-SMOTE, concluding that hybrid approaches are more consistent than individual methods and that performance also depends on the proportion of synthetic samples generated. In turn, [23] compares traditional oversampling techniques with CGAN and CTGAN in medical data with imbalance ratios of 6.8, 9, and 19, finding that GAN-BASED variants achieve better performance than several classic techniques in terms of AUC and F1-score. Finally, [24] shows that even the proportion between synthetic samples from SMOTE and GAN can be optimized by genetic algorithms, which indicates that the success of hybrids does not depend only on the generative model but also on the balance between diversity, realism, and the number of artificial samples. This line of evidence methodologically supports the SMOTE-NC-CTGAN approach evaluated in this study, which is inspired by [12] but differs from it in three aspects.

First, SMOTified-GAN replaces the random noise of the generator with SMOTE samples, whereas here SMOTE-NC only expands the minority class up to 50% of the majority class, and CTGAN is then trained on this expanded set. Second, the interpolated records are discarded afterwards, so the final set contains only real and CTGAN records. Third, SMOTE-NC and the conditional generator of CTGAN [37] handle mixed numerical and categorical variables natively, whereas [12] and [22] used standard GANs on mainly continuous features.

E. Classification Models in Imbalanced Contexts

In addition to the generation method, the literature shows that the final performance depends on the classifier used. The article [1] shows that, in credit card fraud, ensemble and boosting models especially benefit from balancing, and that XGBoost can outperform other classifiers when combined with appropriate oversampling techniques. Similarly, [19] reports that XGBoost, Logistic Regression, and Gradient Boosting had comparable behaviors in zygosity prediction, although XGBoost was ultimately selected for its slight advantage and better subsequent interpretability by SHAP. In cybersecurity, [17] also shows that XGBoost combined with re-sampling techniques achieves a very high balance between sensitivity and specificity, supporting its choice in imbalanced tabular classification contexts.

On the other hand, [25] demonstrates that balancing can also enhance explainable ensemble models in cardiovascular prediction, while [1] and [17] reinforce the idea that tree-based and boosting models are usually more robust than simple linear alternatives when the dataset is heterogeneous or highly imbalanced. Taken together, this evidence justifies the use of XGBoost and CatBoost as the main classifiers in the experimental phase of this study, leaving Logistic Regression as an auxiliary tool for the selection of hyperparameters during synthetic generation.

F. Metrics to Evaluate and Verify Synthetic Data

An important aspect of recent research is that the evaluation of synthetic data should not focus only on the final performance of the classifier. The study [26] explains that, in imbalanced problems, metrics such as precision, recall, and F1-score are usually more useful than accuracy to analyze behavior about the minority class. In addition, mentions that the metric used should depend on the objective of each problem.

Similarly, the work [8] compares CTGAN, CopulaGAN, VAE, and SMOTE in anonymous traffic classification. It points out that it is not enough to only measure predictive performance, but it is also necessary to evaluate the quality of the data generated. In that study, SMOTE managed to increase the F1-score values by about 7.5% compared to the control group, while LightGBM reached an F1-score of 90.3% over eight types of applications. This shows that the final performance changes depending on the generative method and the classifier used subsequently.

In addition, the study [10] mentions that the evaluation should include other aspects such as numerical and categorical fidelity, the preservation of correlations, and the similarity between real and synthetic data. On the other hand, [27] reports that AE-CTGAN obtained better results than an autoencoder combined with a conventional GAN, reaching an average accuracy of 0.9929 for BMR in bladder cancer and 0.9748 for menopausal status in breast cancer. It also significantly decreased the distance between real and synthetic samples.

Finally, [28] demonstrates that synthetic data can also improve generalizability in medical vision tasks. Through the use of conditional StyleGAN, the mAP increased from 0.86 to 0.93, while the recall in flat or depressed lesions increased from 0.72 to 0.87. These results suggest that the evaluation of synthetic data should not focus only on statistical fidelity but also on the real utility and generalization capacity they bring to the model.

G. Research Gap

Despite the advances made in the treatment of class imbalance, there are still some important limitations within the current literature that this study seeks to address:

1) Homogeneity of the approaches compared: Many investigations tend to compare methods belonging to the same family of algorithms, such as variants of SMOTE or different GAN architectures. This makes it more difficult to analyze how different generative approaches respond to the same scenario.

2) Lack of a dual evaluation: There are still few studies that jointly analyze the statistical fidelity of synthetic data and its predictive utility, although both dimensions do not always exhibit the same behavior.

3) Little evidence in extreme scenarios: There is still little evidence on the performance of these techniques when the imbalance exceeds critical ratios greater than 20:1, a context where several interpolation methods tend to lose effectiveness.

In this context, the present research seeks to cover these limitations through a direct comparison between four different approaches: SMOTE-NC as a traditional interpolation method, CTGAN based on deep adversarial learning, TVAE through variational modeling, and a hybrid proposal called SMOTE-NC-CTGAN. In this way, the behavior of these models is evaluated using health and finance datasets, especially in scenarios with high levels of data scarcity.

III. Methodology

This research follows a quasi-experimental and empirical experimental design, developed through a six-stage pipeline. The objective is to evaluate how generative techniques influence the performance of gradient boosting classifiers (Fig. 1).

A. Datasets Presentation

For the study, five datasets from the UC Irvine Machine Learning Repository [29] were used, selected to present different levels of class imbalance and different degrees of complexity. The sets mainly belong to the areas of health and finance, in addition to containing numerical and categorical variables. Likewise, the imbalance ratios range from moderate scenarios to extreme cases close to 29.99:1, allowing the behavior of the techniques to be analyzed under different conditions.

It is important to note that, although all data sets were obtained through the UCI repository, there are differences in their provenance. Datasets 2, 4, and 5 correspond to sets directly integrated into the repository, while datasets 1 and 3 are curated versions of external sources. In particular, dataset 1 originally came from the Behavioral Risk Factor Surveillance System of the Centers for Disease Control and Prevention (CDC), and dataset 3 from ClinicalTrials.gov, both of which were later structured and adapted by UCI for use in machine learning tasks.

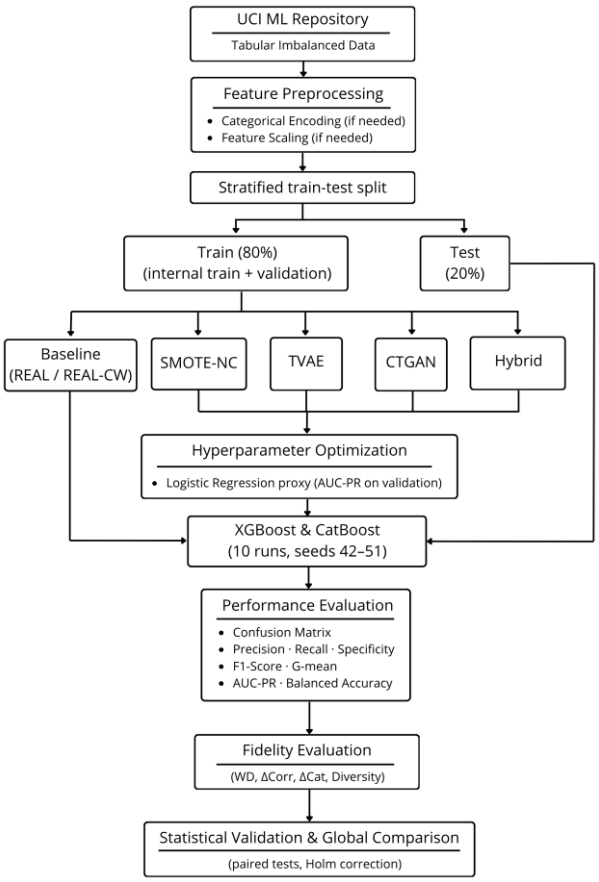


Fig. 1. Methodological flow.

TABLE I. Dataset Distribution Table

Dataset	Name	Domain	Samples	Features	Numerical	Categorical	Ratio	Minority n (%)	Ref.
DS1	Diabetes Health Indicators	Healthcare	253,680	21	5	16	6.18	35,346 (13.9%)	[30], [31]
DS2	Thyroid Cancer Recurrence	Healthcare	383	16	1	15	2.55	108 (28.2%)	[32]
DS3	AIDS Clinical Trial (175)	Healthcare	2,139	23	9	14	3.1	521 (24.4%)	[33], [34]
DS4	Credit Card Default	Finance	30,000	23	14	9	3.5	6,636 (22.1%)	[35]
DS5	Taiwanese Bankruptcy	Finance	6,819	95	93	2	29.99	220 (3.2%)	[36]

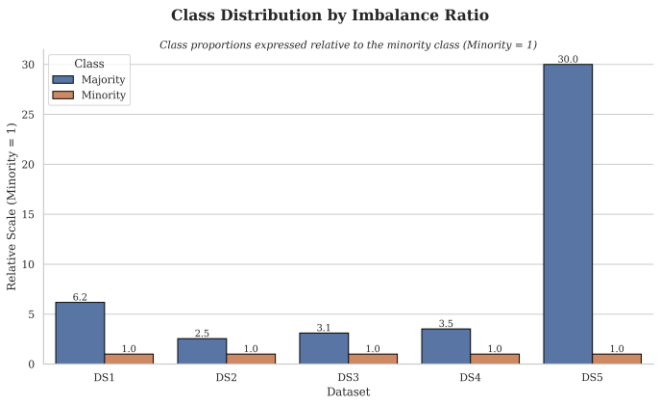


Fig. 2. Distribution of classes by dataset.

It should be noted that the DS5 (Taiwanese Bankruptcy) constitutes the "stress case" of this research, due to its extreme imbalance ratio and its high dimensionality (95 features), factors that impose a critical challenge to the generalizability of generative models.

Imbalance ratio and dataset size are not independent in this benchmark: DS2 and DS5 have the smallest minority classes (Fig. 2). Table I therefore reports the size of the minority class, which is considered in the interpretation (Section V).

B. Data Pre-Processing

Although the datasets come from consolidated sources, a standardized transformation protocol was applied to ensure convergence of the generative models. Both transformations were fitted exclusively on the training partition and then applied, without refitting, to the test partition:

1) Categorical coding: Label Encoding was used to transform nominal variables into integer representations, guaranteeing compatibility with the *Gumbel-Softmax* activations of the CTGAN and TVAE architectures.

2) Numerical standardization: Continuous variables were normalized using StandardScaler, as this facilitates the learning process in neural networks and improves the behavior of distance-based methods, such as SMOTE-NC. CTGAN and TVAE were trained on the variables in their original scale, since they apply their own mode-specific normalization [37]; their outputs were then transformed with the same scaler before being used by the classifiers.

C. Data Division

Each dataset was divided into 80% for training and 20% for testing by stratified sampling (seed 42). The test partition was isolated before any transformation and used only once, for the final evaluation, which prevents data leakage. The training partition was further divided (80/20, stratified) into an internal training subset, the only real data used by the generators and by all classifiers, including the baseline, and a validation subset used only for hyperparameter selection.

D. Generation of Synthetic Data

Four strategies representing different approaches to treating class imbalance were used in this research: a traditional technique (SMOTE-NC), two modern generative models (CTGAN and TVAE [37]), and a hybrid proposal called SMOTE-NC-CTGAN. This allowed us to compare methods with different ways of generating synthetic data.

For all methods, only the minority class of the internal training subset was used as the source, and $M - m$ records were generated to reach a 1:1 ratio (M and m: majority and minority sizes). SMOTE-NC interpolates numerical variables between minority neighbours and assigns the most frequent category among them; CTGAN and TVAE were trained only on minority records. The hybrid approach follows Algorithm 1.

Algorithm 1. SMOTE-NC-CTGAN hybrid generation

Input: internal training subset (X, y) with M majority and m minority records; densification floor $\alpha = 0.5$; k selected for SMOTE-NC; CTGAN hyperparameters θ; scaler S fitted on the training partition.

1: Apply SMOTE-NC with k to the scaled data until the minority class reaches $\lceil \alpha \cdot M \rceil$ records (real and interpolated).

2: Map the expanded minority set back to the original scale with S^{-1}.

3: Train CTGAN(θ) exclusively on the expanded minority set, declaring the categorical variables as discrete columns.

4: Sample $M - m$ synthetic records from the trained CTGAN.

5: Discard the interpolated SMOTE-NC records; the final set is formed by the original real records and the $M - m$ CTGAN records (1:1 ratio).

6: Scale the numerical variables with S before training the classifiers.

Output: balanced training set.

The floor $\alpha = 0.5$ gives CTGAN a larger minority sample while limiting the share of interpolated records; it was fixed a priori and not tuned.

The hyperparameters of the generators were selected per dataset by grid search on the internal train/validation split: $k \in \{3, 5, 7\}$ for SMOTE-NC; (epochs, batch size) $\in \{(300, 500), (500, 500), (500, 1000)\}$ with pac = 10 for CTGAN and the CTGAN stage of the hybrid; and (epochs, batch size, embedding dimension) $\in \{(300, 256, 128), (500, 256, 128), (500, 512, 256)\}$ for TVAE. Each candidate was scored by the AUC-PR on the validation subset of a Logistic Regression proxy, chosen for its low cost and its sensitivity to the training distribution; the selected values are listed in Table VI (Appendix).

E. Classifier Models

Once the balanced datasets were generated, the XGBoost and CatBoost classifiers were trained with a fixed configuration (300 trees, learning rate 0.05, depth 6, log-loss; XGBoost also with 0.8 row and column subsampling) and the default threshold of 0.5, without tuning, so that the differences can be attributed to the training data. Two references were trained on the same real data: REAL, without treatment, and REAL-CW, which weights the minority class by the imbalance ratio (cost-sensitive baseline).

Each experiment was repeated in 10 runs (seeds 42 to 51); each run regenerates all synthetic data and retrains the classifiers, while the train/test partition is fixed. The results are presented as averages accompanied by their Standard Deviation.

F. Fidelity Evaluation and Evaluation Metrics

For this research, an evaluation divided into two parts was proposed, since it is not only important that the synthetic data resemble the originals, but also to check if they really help the performance of the models. In the statistical fidelity part, the Wasserstein Distance was mainly used because it allows comparison of how similar the distributions of the real data are to the synthetic ones. In addition, this metric is based on the theory of optimal transport, so it helps to observe whether the generator model manages to preserve the original structure of the data even when there is high dimensionality.

Likewise, the analysis also considers the Correlation Difference and the Categorical Difference because both help to verify whether the relationships between variables and the proportions of the categories remain similar with respect to the original dataset. Likewise, the Diversity Score allows evaluation of whether the model is capable of generating varied data and not simply repeating very similar records. It is compared with the diversity of the real minority records: values far above it indicate records dispersed beyond the real data.

On the other hand, to measure predictive utility, the Area Under the Precision-Recall Curve (AUC-PR) was taken as the main metric, mainly because it offers a better analysis in scenarios where there is a severe imbalance of classes: its baseline equals the prevalence of the minority class, and it does not depend on a threshold [38]. In addition, this analysis was complemented by Precision and F1-Score, since they allow observing the balance between correctly detecting the minority class and avoiding unnecessary false alarms.

Recall was also considered because it allows us to measure more directly how many cases of the minority class are recovered correctly by the model, and Specificity and the G-mean [39] were added to quantify the cost on the majority class. Meanwhile, the Balanced Accuracy helps maintain a more balanced assessment considering the performance of both classes. Finally, the confusion matrices allowed these results to be interpreted in a more practical way.

Additionally, a hypothesis testing protocol was incorporated to determine if the differences observed between the methods are attributable to the synthesis procedure or to random variability between runs. Since all conditions share runs, seeds, and test partition, the design is paired. For each family of four comparisons (synthetic methods versus REAL, same dataset, classifier and metric), p-values were corrected with Holm-Bonferroni, and the effect size (d_z) and the 95% confidence interval are reported; REAL-CW is tested separately.

Normality of the paired differences was checked with Shapiro-Wilk; when it was rejected ($p<0.05$), which is frequent for Recall or Precision in small test sets, the Wilcoxon signed-rank test was used instead of the t-test. Independence holds at the level of runs, although all runs share the same test partition, so the inference refers to the variability of the procedure. Non-significant results indicate insufficient evidence, not absence of effect.

Finally, all comparisons keep the classifier constant; Table III and the figures show CatBoost, and Table VII (Appendix) shows XGBoost.

IV. Results

Below are the main findings obtained after executing the experimental pipeline:

A. Statistical Fidelity Analysis (Quality Metrics)

In this first stage, the ability of generative models to replicate the distribution of the minority class is evaluated. This phase is important because if the synthetic data does not maintain adequate statistical fidelity, the classifier could lose the ability to correctly generalize to real data.

The results are presented in Table II, where the performance of the four paradigms is compared using different quality metrics. Among them is the Wasserstein Distance (WD), used to measure the similarity between real and synthetic data. In addition, the Correlation Difference (Corr) and the Categorical Difference (CAT) were used to verify whether the relationships between variables and the categorical proportions are adequately preserved. Finally, the Diversity Score (DIV) allowed evaluation of whether the model generates varied samples and avoids producing repetitive data that may affect learning.

The analysis of Table II shows different behaviors according to the characteristics of each dataset:

Domain of interpolation in all scenarios: SMOTE-NC obtains the lowest Wasserstein distance in the five datasets. In DS1 (Diabetes Health Indicators), with a ratio of 6.18:1, it reaches 0.0429 ± 0.0005, compared with 0.1652 for CTGAN and 0.1471 for TVAE, which indicates that interpolation preserves the statistical structure well when there is a sufficient amount of real data.

Response in categorical datasets: An interesting case appears in the DS2 (Thyroid Cancer). Although it has the lowest ratio (2.55:1), the presence of many categorical variables represents a difficulty for several traditional methods. In this scenario, HYBRID achieves a better adaptation to the nominal proportions, reaching the smallest Categorical Difference (0.1269 ± 0.0279), thus surpassing SMOTE-NC (0.2887) and TVAE (0.3208).

Greater diversity in extreme scenarios: In the DS5, with a ratio of 29.99:1, there is a more marked difference between fidelity and diversity. SMOTE-NC keeps a low WD (0.0913) and a diversity (9.24) close to that of the real records (12.11), whereas CTGAN and HYBRID reach 42.55 and 46.20, together with the highest WD of the study (1.1981 and 1.0716). This diversity mostly reflects records dispersed outside the real minority class, consistent with the only 141 minority records available in 95 dimensions.

To complement this analysis, Fig. 3 shows the evaluation of fidelity and diversity corresponding to the base scenario DS1.

TABLE II. GLOBAL SUMMARY OF STATISTICAL FIDELITY METRICS (MEAN ± SD OVER 10 RUNS)

Dataset	Ratio	Method	WD (↓)	ΔCorr* (↓)	ΔCat (↓)	Div. (≈ real)
DS1	6.18	SMOTE-NC	0.0429 (± 0.0005)	0.0087 (± 0.0008)	0.0548 (± 0.0006)	2.8363 (± 0.0277)
		CTGAN	0.1652 (± 0.0449)	0.0726 (± 0.0265)	0.1453 (± 0.0178)	3.0061 (± 0.2121)
		TVAE	0.1471 (± 0.0370)	0.0635 (± 0.0114)	0.1168 (± 0.0168)	2.4460 (± 0.1620)
		HYBRID	0.1449 (± 0.0309)	0.0635 (± 0.0184)	0.1265 (± 0.0344)	2.8552 (± 0.1382)
		Real minority	—	—	—	2.8756
DS2	2.55	SMOTE-NC	0.1278 (± 0.0476)	-	0.2887 (± 0.0250)	1.0809 (± 0.0565)
		CTGAN	0.5730 (± 0.1989)	-	0.1982 (± 0.0576)	1.7503 (± 0.1011)
		TVAE	0.1347 (± 0.0422)	-	0.3208 (± 0.0317)	1.0814 (± 0.0358)
		HYBRID	0.7813 (± 0.2234)	-	0.1269 (± 0.0279)	1.2114 (± 0.1222)
		Real minority	—	—	—	1.1628
DS3	3.10	SMOTE-NC	0.0958 (± 0.0077)	0.0355 (± 0.0028)	0.1659 (± 0.0080)	3.5990 (± 0.0348)
		CTGAN	0.6533 (± 0.0976)	0.1335 (± 0.0100)	0.1195 (± 0.0146)	5.0332 (± 0.2063)
		TVAE	0.2305 (± 0.0232)	0.0751 (± 0.0069)	0.2188 (± 0.0199)	3.4832 (± 0.0438)
		HYBRID	0.5045 (± 0.0975)	0.1311 (± 0.0125)	0.1273 (± 0.0109)	4.9659 (± 0.2123)
		Real minority	—	—	—	4.0682
DS4	3.52	SMOTE-NC	0.0281 (± 0.0035)	0.0170 (± 0.0027)	0.0621 (± 0.0028)	3.6468 (± 0.1274)
		CTGAN	0.1358 (± 0.0132)	0.1981 (± 0.0079)	0.3000 (± 0.0282)	4.4143 (± 0.1854)
		TVAE	0.0901 (± 0.0083)	0.0701 (± 0.0024)	0.1052 (± 0.0256)	3.5753 (± 0.1314)
		HYBRID	0.1159 (± 0.0138)	0.0909 (± 0.0059)	0.2758 (± 0.0401)	3.8830 (± 0.1048)
		Real minority	—	—	—	3.8432
DS5	29.99	SMOTE-NC	0.0913 (± 0.0021)	0.0303 (± 0.0015)	0.0146 (± 0.0011)	9.2412 (± 0.1190)
		CTGAN	1.1981 (± 0.0699)	0.1523 (± 0.0004)	0.0937 (± 0.0032)	42.5466 (± 1.4128)
		TVAE	0.2637 (± 0.0037)	0.1177 (± 0.0023)	0.0213 (± 0.0001)	5.4916 (± 0.1263)
		HYBRID	1.0716 (± 0.0569)	0.1482 (± 0.0058)	0.1144 (± 0.0110)	46.2043 (± 1.0787)
		Real minority	—	—	—	12.1129

(↓) indicates lower values are better; (≈ real) indicates that values closest to the diversity of the real minority records are preferable.

WD: Wasserstein Distance; ΔCorr: Correlation Difference; ΔCat: Categorical Difference; Div: Mean Pairwise Distance; Real minority: diversity of the real minority records of the internal training subset, used as reference.

*Note: ΔCorr is not applicable (-) for DS2 because it contains only one numerical feature (Table I), making pairwise correlation uncalculable.

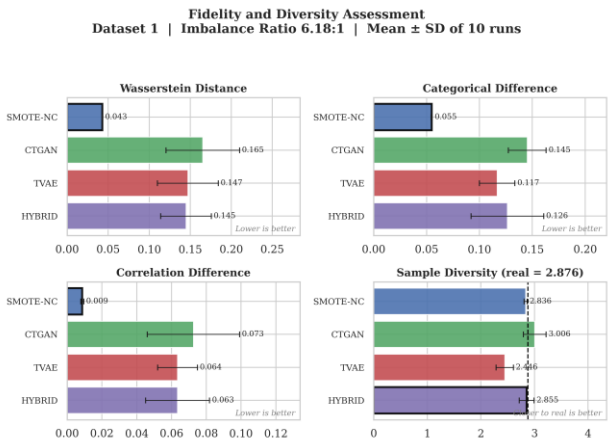


Fig. 3. Dataset fidelity and diversity evaluation 1.

Fig. 3 shows that in the Correlation Difference panel all methods stay at low values, with SMOTE-NC presenting the lowest error and an almost zero variance between runs. The Sample Diversity panel includes the diversity of the real minority records as a reference: in DS1, SMOTE-NC and HYBRID remain closest to it, CTGAN slightly exceeds it, and TVAE generates less varied records.

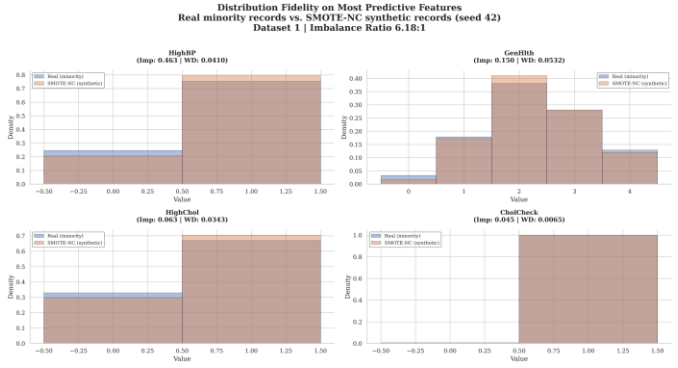


Fig. 4. Distribution fidelity of SMOTE-NC in the most predictive features of Dataset 1 (real minority vs. synthetic records).

Finally, Fig. 4 compares the real minority records of DS1 with the SMOTE-NC synthetic records (seed 42) in the four most important features (XGBoost importance on the internal training subset).

Fig. 4 shows real minority records in blue and synthetic records in orange. All distances are low (WD between 0.0065 and 0.0532), consistent with Table II, and three patterns are identified:

1) Almost perfect mimicry: In CholCheck (WD = 0.0065), 99.34% of the real and 99.99% of the synthetic records take the value 1, so the distribution is practically reproduced.

2) Reinforcement of the dominant category: In HighBP (Imp: 0.463), the most relevant feature, the share of the value 1 rises from 75.3% to 79.4% (WD = 0.0410), and in HighChol from 67.1% to 70.5%. This occurs because SMOTE-NC assigns to each synthetic record the most frequent category among its neighbours, which slightly inflates the dominant categories.

3) Ordinal smoothing: GenHlth shows the largest distance (WD = 0.0532): the central category gains weight (38.1% to 41.0%) and the extreme ones lose it (e.g., 3.3% to 1.7% in category 0), so rare values are under-represented, although the ordinal structure of the variable is preserved.

B. Evaluation of Predictive Performance (Utility Analysis)

After validating the statistical fidelity, we proceed to evaluate the predictive utility of synthetic data, understood as its ability to improve the training of classifiers in scenarios with severe imbalance. This analysis seeks to determine whether the samples generated help to better identify the minority class without significantly affecting the overall performance of the system.

The consolidated results with CatBoost are presented in Table III, and those with XGBoost in Table VII (Appendix). These values were obtained over 10 runs that regenerate the synthetic data, and they include the cost-sensitive baseline REAL-CW.

1) Ranking resilience analysis (AUC-PR): The analysis of Table III allows us to identify different behaviors depending on the level of imbalance present in each dataset:

a) Behavior in moderate ratios (DS1 to DS4): In scenarios with ratios between 2.55:1 and 6.18:1, no synthesis method improves the AUC-PR of the model trained with real data. In DS2 (Thyroid Cancer), all conditions are practically saturated (REAL: 0.9821; CTGAN: 0.9859), whereas in DS1 and DS4 all synthetic methods reduce AUC-PR significantly, with the largest losses for the deep models (e.g., DS1: 0.4242 with REAL versus 0.3482 with CTGAN and 0.3222 with TVAE).

b) Performance at the critical limit (DS5): In the scenario with a 29.99:1 ratio, the effect depends on the classifier. With XGBoost, HYBRID (0.5252 ± 0.0222) and CTGAN (0.5137) significantly exceed the baseline (0.4872; p=0.017 and p=0.021 after Holm correction); with CatBoost, no method exceeds the baseline (0.5131), and HYBRID obtains 0.4902. Moreover, the best XGBoost result remains at the level of the CatBoost baseline, so the improvement is relative to a weaker classifier and should not be interpreted as a global superiority of the hybrid method (Section IV-D).

TABLE III. PREDICTIVE PERFORMANCE ON THE TEST PARTITION WITH CATBOOST (MEAN ± SD OVER 10 RUNS)

Dataset	Ratio	Method	AUC-PR	Recall	Precision	F1-Score	Specificity	G-mean	Bal. Acc.
DS1	6.18	REAL	0.4242 ± 0.0005	0.1617 ± 0.0016	0.5581 ± 0.0030	0.2508 ± 0.0021	0.9793 ± 0.0002	0.3980 ± 0.0020	0.5705 ± 0.0008
		REAL-CW	0.4237 ± 0.0003	0.7938 ± 0.0008	0.3079 ± 0.0002	0.4437 ± 0.0002	0.7112 ± 0.0003	0.7513 ± 0.0003	0.7525 ± 0.0003
		SMOTE-NC	0.4182 ± 0.0003	0.2549 ± 0.0017	0.5021 ± 0.0019	0.3382 ± 0.0017	0.9591 ± 0.0004	0.4945 ± 0.0016	0.6070 ± 0.0008
		CTGAN	0.3482 ± 0.0079	0.5835 ± 0.0249	0.3256 ± 0.0084	0.4176 ± 0.0057	0.8040 ± 0.0145	0.6847 ± 0.0093	0.6937 ± 0.0065
		TVAE	0.3222 ± 0.0084	0.4582 ± 0.0436	0.3257 ± 0.0090	0.3797 ± 0.0139	0.8460 ± 0.0177	0.6216 ± 0.0243	0.6521 ± 0.0135
		HYBRID	0.4138 ± 0.0024	0.1781 ± 0.0032	0.5320 ± 0.0049	0.2669 ± 0.0037	0.9746 ± 0.0006	0.4166 ± 0.0037	0.5764 ± 0.0014
DS2	2.55	REAL	0.9821 ± 0.0030	0.9091 ± 0.0000	1.0000 ± 0.0000	0.9524 ± 0.0000	1.0000 ± 0.0000	0.9535 ± 0.0000	0.9545 ± 0.0000
		REAL-CW	0.9819 ± 0.0042	0.9091 ± 0.0000	1.0000 ± 0.0000	0.9524 ± 0.0000	1.0000 ± 0.0000	0.9535 ± 0.0000	0.9545 ± 0.0000
		SMOTE-NC	0.9807 ± 0.0026	0.9091 ± 0.0000	1.0000 ± 0.0000	0.9524 ± 0.0000	1.0000 ± 0.0000	0.9535 ± 0.0000	0.9545 ± 0.0000
		CTGAN	0.9859 ± 0.0031	0.9136 ± 0.0144	0.9861 ± 0.0309	0.9483 ± 0.0176	0.9945 ± 0.0123	0.9532 ± 0.0102	0.9541 ± 0.0102
		TVAE	0.9803 ± 0.0033	0.9091 ± 0.0000	1.0000 ± 0.0000	0.9524 ± 0.0000	1.0000 ± 0.0000	0.9535 ± 0.0000	0.9545 ± 0.0000
		HYBRID	0.9857 ± 0.0058	0.9136 ± 0.0258	0.9907 ± 0.0196	0.9503 ± 0.0137	0.9964 ± 0.0077	0.9540 ± 0.0126	0.9550 ± 0.0120
DS3	3.10	REAL	0.8717 ± 0.0025	0.7442 ± 0.0081	0.8090 ± 0.0178	0.7752 ± 0.0110	0.9435 ± 0.0062	0.8380 ± 0.0062	0.8439 ± 0.0060
		REAL-CW	0.8683 ± 0.0053	0.7913 ± 0.0091	0.7609 ± 0.0158	0.7757 ± 0.0094	0.9201 ± 0.0070	0.8533 ± 0.0056	0.8557 ± 0.0054
		SMOTE-NC	0.8692 ± 0.0039	0.7885 ± 0.0111	0.7826 ± 0.0110	0.7854 ± 0.0062	0.9296 ± 0.0050	0.8561 ± 0.0052	0.8590 ± 0.0047
		CTGAN	0.8666 ± 0.0092	0.7644 ± 0.0130	0.7662 ± 0.0166	0.7652 ± 0.0114	0.9250 ± 0.0071	0.8409 ± 0.0077	0.8447 ± 0.0073

		TVAE	0.8652	±	0.7856	±	0.7950	±	0.7902	±	0.9349	±	0.8570	±	0.8602	±
			0.0050		0.0111		0.0171		0.0118		0.0066		0.0074		0.0072	
		HYBRID	0.8677	±	0.7481	±	0.7789	±	0.7631	±	0.9318	±	0.8348	±	0.8399	±
			0.0070		0.0174		0.0095		0.0100		0.0040		0.0091		0.0082	
DS4	3.52	REAL	0.5593	±	0.3648	±	0.6613	±	0.4702	±	0.9469	±	0.5877	±	0.6559	±
			0.0021		0.0025		0.0025		0.0024		0.0006		0.0020		0.0012	
		REAL-CW	0.5598	±	0.6130	±	0.4825	±	0.5400	±	0.8133	±	0.7061	±	0.7131	±
			0.0024		0.0026		0.0019		0.0016		0.0016		0.0013		0.0012	
		SMOTE-NC	0.5515	±	0.4669	±	0.5638	±	0.5108	±	0.8974	±	0.6473	±	0.6822	±
			0.0018		0.0050		0.0054		0.0039		0.0024		0.0032		0.0024	
		CTGAN	0.5300	±	0.3949	±	0.6318	±	0.4860	±	0.9346	±	0.6075	±	0.6648	±
			0.0036		0.0041		0.0057		0.0031		0.0019		0.0028		0.0017	
		TVAE	0.5250	±	0.4352	±	0.5771	±	0.4962	±	0.9094	±	0.6291	±	0.6723	±
			0.0030		0.0053		0.0076		0.0041		0.0032		0.0034		0.0023	
		HYBRID	0.5328	±	0.3958	±	0.6226	±	0.4839	±	0.9318	±	0.6073	±	0.6638	±
			0.0063		0.0047		0.0120		0.0048		0.0037		0.0034		0.0025	
DS5	29.99	REAL	0.5131	±	0.2409	±	0.6157	±	0.3456	±	0.9950	±	0.4885	±	0.6180	±
			0.0245		0.0342		0.0540		0.0415		0.0008		0.0351		0.0171	
		REAL-CW	0.4727	±	0.6023	±	0.4504	±	0.5152	±	0.9755	±	0.7664	±	0.7889	±
			0.0286		0.0193		0.0187		0.0163		0.0018		0.0123		0.0096	
		SMOTE-NC	0.5045	±	0.6273	±	0.4168	±	0.5007	±	0.9707	±	0.7803	±	0.7990	±
			0.0264		0.0117		0.0187		0.0160		0.0020		0.0077		0.0064	
		CTGAN	0.5119	±	0.2455	±	0.6093	±	0.3484	±	0.9947	±	0.4929	±	0.6201	±
			0.0268		0.0383		0.0575		0.0404		0.0012		0.0366		0.0189	
		TVAE	0.4472	±	0.3955	±	0.5790	±	0.4695	±	0.9905	±	0.6248	±	0.6930	±
			0.0220		0.0469		0.0433		0.0461		0.0010		0.0376		0.0236	
		HYBRID	0.4902	±	0.2409	±	0.6160	±	0.3453	±	0.9948	±	0.4889	±	0.6179	±
			0.0514		0.0267		0.0901		0.0367		0.0017		0.0273		0.0135	

REAL: real data without treatment; REAL-CW: real data with class weights (cost-sensitive baseline). The results with XGBoost are reported in Table VII (Appendix)

To complement this analysis, Fig. 5 shows the evolution of the performance of the different methods as the level of imbalance increases.

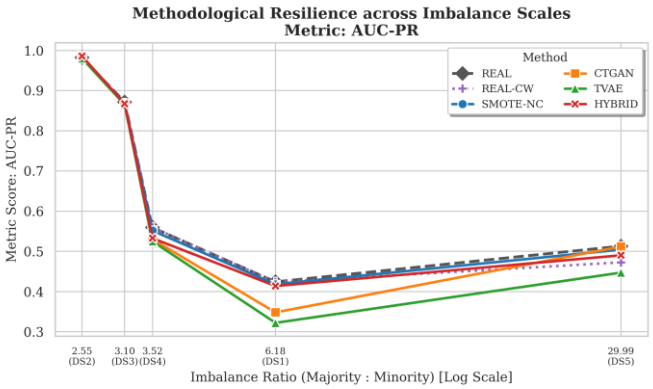


Fig. 5. Methodological resilience through imbalance scales (Metric: AUC-PR).

Fig. 5 (CatBoost) shows that the AUC-PR varies more with the dataset than with the generation method; the only significant improvements appear with XGBoost in DS5 (Table V). Fig. 6 details the differences in DS5 for both classifiers; given the small number of positive records in the test partition of this dataset (44), these differences are sensitive to a few cases.

2) Stability and fair evaluation (balanced accuracy): Because traditional Accuracy can be misleading in imbalanced scenarios, Balanced Accuracy was used to evaluate performance by considering both classes in a more balanced way.

a) Impact of balancing: Balanced Accuracy increases significantly in 21 of the 40 comparisons. In DS1, for example, CTGAN increases it from 0.5705 to 0.6937, and the cost-sensitive baseline REAL-CW reaches 0.7525. This shows that the classifier stops mainly favoring the majority class.

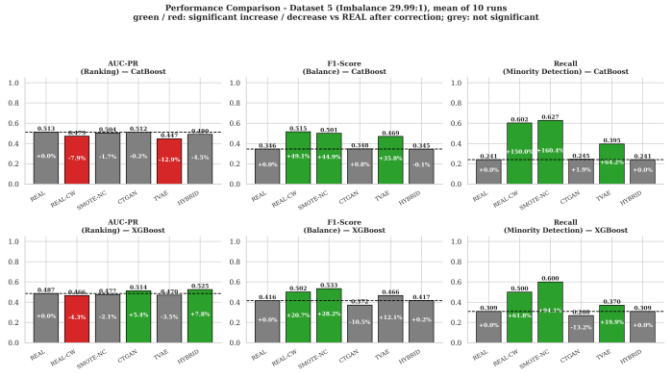


Fig. 6. Relative performance improvements among generative methods for dataset 5.

b) Behavior in the extreme scenario: In DS5, SMOTE-NC (0.7990) and REAL-CW (0.7889) obtain the highest Balanced Accuracy, followed by TVAE (0.6930), while CTGAN (0.6201) and HYBRID (0.6179) remain at the level of the baseline (0.6180).

3) Operational sensitivity and stability (recall and F1-score): The most important impact of data synthesis is observed in the model's ability to detect rare events, evaluated by Recall and F1-Score. According to the results of Table III, the use of synthetic data considerably modifies the behavior of the classifier.

a) Displacement of the decision boundary: In DS1 (Diabetes), the model trained with real data obtains a Recall of only 0.1617 ± 0.0016. By incorporating CTGAN, this value

increases to 0.5835 ± 0.0249, and REAL-CW reaches 0.7938 without any synthetic records. The hybrid approach is the most conservative (0.1781), since its final set contains only CTGAN records generated from a densified minority.

Gain in extreme scenarios (DS5): In the scenario of financial bankruptcy, SMOTE-NC obtains the largest increase in Recall, from 0.2409 to 0.6273 (CatBoost), similar to REAL-CW (0.6023), and both obtain the highest F1-Score (0.5007 and 0.5152 versus 0.3456). This improvement is accompanied by a reduction in precision. CTGAN and HYBRID do not change Recall significantly in this dataset.

As seen in Fig. 7, the F1-Score improves mainly in the scenarios where the baseline detects few positive cases (DS1, DS4 and DS5), and the largest values are obtained by REAL-CW and SMOTE-NC. In DS5, the F1-Score of the CTGAN-based methods also shows the largest variability between runs.

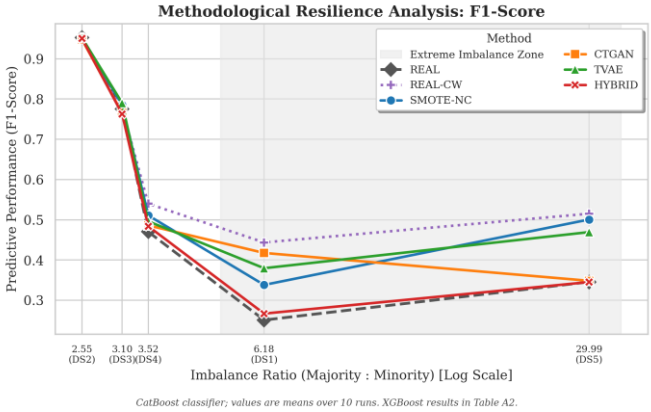


Fig. 7. F1 score across imbalance scales and methodological resilience.

4) Diagnostic quality and trade-off (precision): To complement the analysis, precision was also evaluated, a metric that allows us to measure how reliable the positive predictions made by the model are. In imbalanced scenarios, this metric is important because it helps to observe the cost of increasing sensitivity through synthetic data.

a) Relationship between precision and recall: In DS1 (Diabetes), CTGAN raises Recall to 58.3%, but reduces Precision from 0.5581 (REAL) to 0.3256; that is, about two out of every three positive predictions are false alarms. Across the 40 comparisons, Precision decreases significantly in 25 and Specificity in 27, while the G-mean, which balances Recall and Specificity, increases in 23.

b) Behavior in extreme imbalance (DS5): While SMOTE-NC reduces Precision to 0.4168 to reach a Recall of 0.6273, TVAE (0.5790) and HYBRID (0.6160) maintain a precision close to that of the baseline (0.6157), but with a smaller gain in Recall. The methods therefore move along the precision-recall trade-off rather than improving both metrics at once.

To complement this analysis, Fig. 8 shows the operating point (Recall, Precision) of each condition at the default threshold, together with iso-F1 curves. In all datasets except DS2, the synthetic methods and REAL-CW move the operating point towards higher Recall and lower Precision, and the points remain close to the same iso-F1 curves, which confirms that the main effect is a displacement of the decision threshold rather than a better separation of the classes.

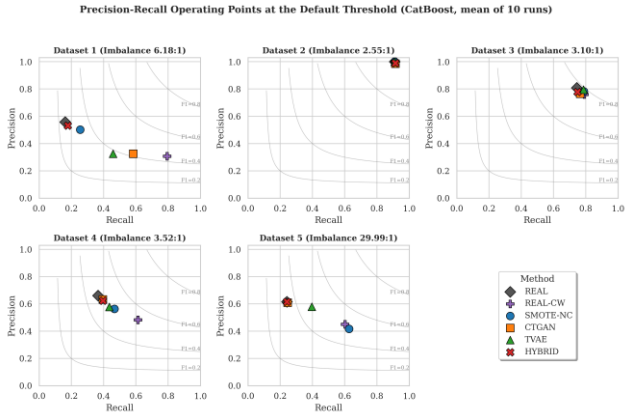


Fig. 8. Precision-Recall operating points at the default threshold (CatBoost).

5) Operational breakdown of positive detections: Fig. 9 to 13 show the mean confusion matrices over the 10 runs (CatBoost) of the baseline and of one representative method per dataset.

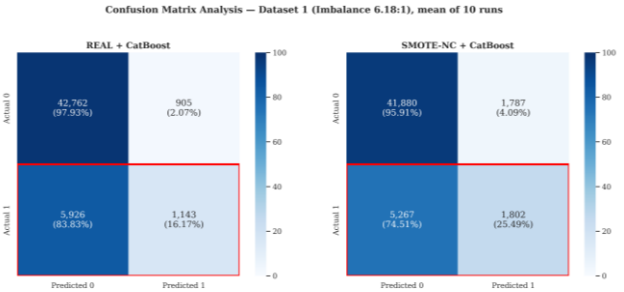


Fig. 9. Confusion matrix analysis: Dataset 1

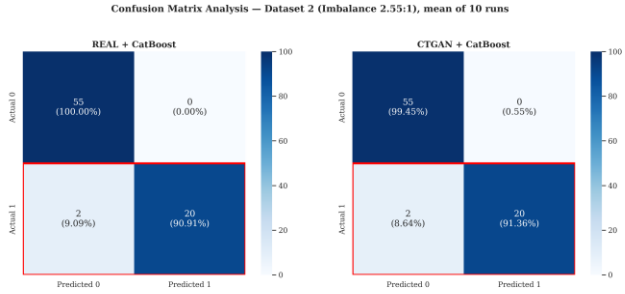


Fig. 10. Confusion matrix analysis: Dataset 2.

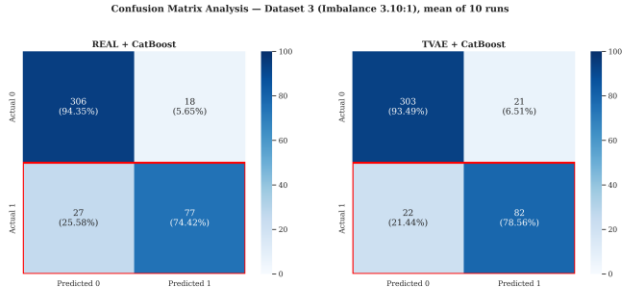


Fig. 11. Confusion matrix analysis: Dataset 3.

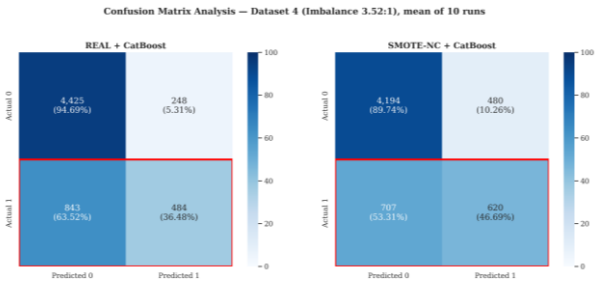


Fig. 12. Confusion matrix analysis: Dataset 4.

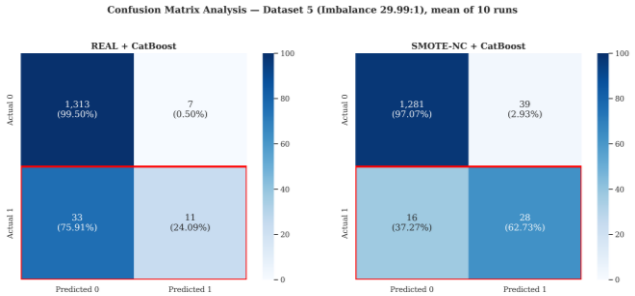


Fig. 13. Confusion matrix analysis: Dataset 5.

Scenario 1: Large-Scale Health Indicators (DS1). With an imbalance ratio of 6.18:1, the base classifier detects, on average, 1,143 of the 7,069 positive cases (16.2%). SMOTE-NC raises this number to 1,802, CTGAN to 4,125, and REAL-CW to 5,611; however, false positives grow from 905 to 8,559 with CTGAN and 12,612 with REAL-CW.

Scenario 2: Moderate Imbalance (DS2, DS3 and DS4). In DS2 (Thyroid Cancer), all conditions detect about 20 of the 22 positive cases, so synthesis does not disturb an already saturated learning. In DS3 (AIDS), SMOTE-NC and TVAE raise the detections from 77.4 to 82.0 and 81.7 of 104 cases. The largest impact of this group occurs in DS4 (Credit Default), where SMOTE-NC increases the detections from 484 to 620 of 1,327 cases, and REAL-CW to 813.

Scenario 3: Behavior at the Operational Limit (DS5). With a ratio of 29.99:1, the base model detects, on average, 10.6 of the 44 companies in bankruptcy. SMOTE-NC (27.6) and REAL-CW (26.5) more than double this number, whereas CTGAN (10.8) and HYBRID (10.6) remain at the level of the baseline. Given the small number of positive cases, each detected company represents 2.3 points of Recall.

C. Trade-Off Analysis: Statistical Fidelity Vs. Predictive Utility

Fig. 14 relates the fidelity error (Wasserstein distance) of each method to its AUC-PR (CatBoost); the dotted lines mark the medians of both axes. SMOTE-NC occupies the low-error side in all datasets, while the CTGAN-based methods move to the right, especially in DS2, DS3, and DS5.

The vertical position depends mainly on the dataset rather than on the method, and the largest fidelity errors (CTGAN and HYBRID in DS5) do not correspond to the largest losses of AUC-PR, which occur in DS1 and DS4 with moderate errors. Therefore, fidelity alone is not a sufficient predictor of utility, which reinforces the need for the dual evaluation.

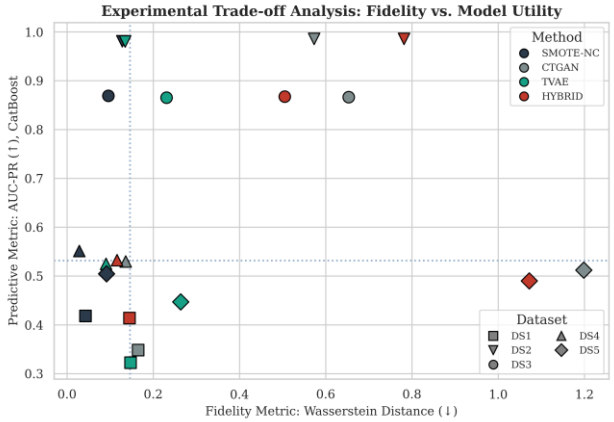


Fig. 14. Trade-off: fidelity vs. model utility.

D. Statistical Validation of Results

Tables II and III present averages and standard deviations over 10 runs, which describe the stability of each method but do not allow establishing whether the differences observed are attributable to the synthesis procedure or to random variability. To resolve this issue, the hypothesis testing protocol described in Section III-F is used. The results are summarized in Table IV.

The analysis of Table IV reveals an asymmetric pattern that substantively qualifies the interpretation of the results:

Few Improvements in AUC-PR. of the 40 Comparisons (five Datasets × Two Classifiers × Four Methods): AUC-PR Decreases Significantly in 19, Does Not Change Significantly in 19, and Increases in Only 2: CTGAN and HYBRID in DS5 with XGBoost. The decreases are concentrated in DS1 and DS4, where the four methods reduce AUC-PR with both classifiers.

Robust and Consistent Improvements in Recall. The picture is reversed when evaluating the Detection of the Minority Class. Of the 40 Comparisons, 27 Show Significant Increases in Recall, none show a Significant Decrease, 8 Are Inconclusive, and 5 Correspond to DS2, where the Metric Is Saturated. The counterpart is precision, Which Decreases in 25 Comparisons. The G-mean Increases in 23, and the F1-Score in 18, with No Significant Decrease in Either Metric.

Dependence on the Classifier in the Extreme Scenario. As shown in Table V, the Sign of the Effect in DS5 depends on the Classifier: under XGBoost, HYBRID exceeds the baseline by +0.0380 (p=0.017 after Holm correction), while under CatBoost it is below by -0.0229 (p=0.640). In addition, the XGBoost baseline (0.4872) is lower than the CatBoost baseline (0.5131), so the improvement obtained with XGBoost brings this classifier to the level of CatBoost rather than above it. A possible explanation of this reversal is discussed in Section V.

On the whole, statistical validation leads to a more accurate reading of the results: data synthesis does not improve the overall ordering of predictions measured by AUC-PR and, in several scenarios, degrades it significantly. Its contribution is concentrated in a reproducible displacement of the decision boundary towards the minority class, which translates into increases in Recall and G-mean of large magnitude, at the cost of Precision and Specificity.

TABLE IV. STATISTICAL VALIDATION: MEAN DIFFERENCE WITH RESPECT TO REAL (Δ), SIGNIFICANCE AND EFFECT SIZE D_Z (N = 10 PAIRED RUNS, SAME CLASSIFIER)

Dataset	Method	AUC-PR		Recall		Precision		G-mean	
		CatBoost	XGBoost	CatBoost	XGBoost	CatBoost	XGBoost	CatBoost	XGBoost
DS1 (6.18)	REAL-CW	-0.0006** (-1.1)	+0.0009** (1.2)	+0.6320*** (337)	+0.6183*** (330)	-0.2502*** (-87)	-0.2414*** (-81)	+0.3534*** (184)	+0.3465*** (200)
	SMOTE-NC	-0.0060*** (-10)	-0.0102*** (-9.3)	+0.0932*** (69)	+0.1702** (68)	-0.0561*** (-15)	-0.0920*** (-18)	+0.0965*** (66)	+0.1576** (79)
	CTGAN	-0.0760*** (-9.6)	-0.0847*** (-12)	+0.4218*** (17)	+0.3921*** (16)	-0.2325*** (-25)	-0.2271*** (-26)	+0.2867*** (31)	+0.2692*** (26)
	TVAE	-0.1020*** (-12)	-0.1029*** (-12)	+0.2964*** (6.9)	+0.2671*** (6.6)	-0.2325*** (-22)	-0.2255*** (-24)	+0.2237*** (9.5)	+0.2040*** (8.6)
	HYBRID	-0.0104** (-3.8)	-0.0076*** (-3.2)	+0.0164*** (4.1)	+0.0118*** (2.5)	-0.0262*** (-5.2)	-0.0221*** (-3.8)	+0.0187*** (4.0)	+0.0133*** (2.5)
DS2 (2.55)	REAL-CW	-0.0002 (-0.0)	-0.0010* (-1.0)	0 (—)	0 (—)	0 (—)	-0.0314* (-1.4)	0 (—)	-0.0063* (-1.4)
	SMOTE-NC	-0.0014 (-0.4)	-0.0003 (-0.2)	0 (—)	0 (—)	0 (—)	-0.0136 (-0.6)	0 (—)	-0.0027 (-0.6)
	CTGAN	+0.0038 (0.8)	-0.0006 (-0.2)	+0.0045 (0.3)	0 (—)	-0.0139 (-0.4)	-0.0265 (-1.2)	-0.0003 (-0.0)	-0.0054 (-1.2)
	TVAE	-0.0018 (-0.4)	-0.0009 (-0.9)	0 (—)	0 (—)	0 (—)	-0.0045 (-0.2)	0 (—)	-0.0009 (-0.2)
	HYBRID	+0.0036 (0.6)	+0.0003 (0.1)	+0.0045 (0.2)	-0.0045 (-0.3)	-0.0093 (-0.5)	-0.0132 (-0.4)	+0.0005 (0.0)	-0.0050 (-0.5)
DS3 (3.10)	REAL-CW	-0.0034 (-0.6)	-0.0030* (-0.8)	+0.0471*** (4.5)	+0.0288*** (2.4)	-0.0481*** (-2.3)	-0.0269*** (-2.7)	+0.0153*** (2.1)	+0.0105** (1.5)
	SMOTE-NC	-0.0025 (-0.5)	-0.0026 (-0.7)	+0.0442*** (2.9)	+0.0250** (1.9)	-0.0264** (-1.6)	-0.0278*** (-2.7)	+0.0181*** (2.3)	+0.0084* (1.2)
	CTGAN	-0.0051 (-0.6)	-0.0089 (-0.8)	+0.0202** (1.6)	+0.0269*** (2.0)	-0.0428*** (-2.2)	-0.0523** (-3.0)	+0.0029 (0.4)	+0.0047 (0.5)
	TVAE	-0.0065** (-1.4)	-0.0114** (-1.6)	+0.0413*** (2.5)	+0.0231* (1.1)	-0.0140 (-0.6)	-0.0365*** (-2.0)	+0.0190** (1.7)	+0.0058 (0.5)
	HYBRID	-0.0040 (-0.5)	-0.0050 (-0.7)	+0.0038 (0.2)	+0.0192* (0.9)	-0.0302** (-1.6)	-0.0349*** (-1.8)	-0.0031 (-0.3)	+0.0041 (0.4)
DS4 (3.52)	REAL-CW	+0.0005 (0.2)	-0.0019 (-0.5)	+0.2482*** (63)	+0.2255*** (42)	-0.1787*** (-53)	-0.1735*** (-40)	+0.1183*** (43)	+0.1079*** (30)
	SMOTE-NC	-0.0078*** (-3.5)	-0.0212*** (-5.4)	+0.1021*** (16)	+0.1039*** (28)	-0.0975*** (-15)	-0.1259*** (-13)	+0.0596*** (13)	+0.0554*** (16)
	CTGAN	-0.0293*** (-6.9)	-0.0127*** (-2.5)	+0.0301*** (6.4)	+0.0170*** (3.2)	-0.0295*** (-4.1)	-0.0208*** (-2.8)	+0.0197*** (5.8)	+0.0110*** (2.5)
	TVAE	-0.0342*** (-8.1)	-0.0264*** (-6.5)	+0.0704*** (12)	+0.0444*** (6.5)	-0.0841*** (-12)	-0.0686*** (-10)	+0.0413*** (11)	+0.0255*** (5.3)
	HYBRID	-0.0265*** (-3.8)	-0.0159*** (-2.1)	+0.0310*** (6.2)	+0.0130*** (2.2)	-0.0387*** (-3.2)	-0.0251*** (-2.0)	+0.0195*** (5.2)	+0.0077*** (1.6)
DS5 (29.99)	REAL-CW	-0.0404* (-0.9)	-0.0210* (-0.7)	+0.3614*** (9.6)	+0.1909*** (3.3)	-0.1652*** (-3.5)	-0.1341*** (-2.2)	+0.2779*** (7.8)	+0.1476*** (3.2)
	SMOTE-NC	-0.0086 (-0.3)	-0.0104 (-0.5)	+0.3864*** (11)	+0.2909*** (6.0)	-0.1988*** (-3.9)	-0.1588*** (-3.1)	+0.2918*** (8.5)	+0.2126*** (5.4)
	CTGAN	-0.0012 (-0.0)	+0.0265* (1.1)	+0.0045 (0.1)	-0.0409 (-0.7)	-0.0063 (-0.1)	-0.0264 (-0.3)	+0.0044 (0.1)	-0.0380 (-0.7)
	TVAE	-0.0659*** (-2.1)	-0.0168 (-0.4)	+0.1545*** (3.3)	+0.0614* (1.0)	-0.0367 (-0.5)	-0.0062 (-0.1)	+0.1364*** (3.5)	+0.0525* (1.0)
	HYBRID	-0.0229 (-0.4)	+0.0380* (1.2)	+0.0000 (0.0)	+0.0000 (0.0)	+0.0003 (0.0)	+0.0060 (0.1)	+0.0004 (0.0)	+0.0004 (0.0)

Cell format: Δ with significance (effect size d_z). *** p<0.001; ** p<0.01; * p<0.05; no mark: not significant. p-values from the paired t-test, or from the Wilcoxon signed-rank test when Shapiro-Wilk rejects the normality of the differences, with Holm correction within each family of four synthetic methods (same dataset, classifier and metric). REAL-CW is tested separately without correction. 0 (—): identical results in all runs (metric saturated). Complete results for all metrics, confidence intervals and uncorrected p-values are available from the corresponding author upon request.

TABLE V. DECOMPOSITION OF THE DS5 SCENARIO (AUC-PR) BY CLASSIFIER (MEAN ± SD OVER 10 RUNS)

Method	CatBoost	Δ vs REAL	p (Holm)	XGBoost	Δ vs REAL	p (Holm)
REAL	0.5131 ± 0.0245	—	—	0.4872 ± 0.0173	—	—
REAL-CW	0.4727 ± 0.0286	-0.0404	0.017†	0.4662 ± 0.0176	-0.0210	0.043†
SMOTE-NC	0.5045 ± 0.0264	-0.0086	0.822	0.4768 ± 0.0101	-0.0104	0.246
CTGAN	0.5119 ± 0.0268	-0.0012	0.922	0.5137 ± 0.0162	+0.0265	0.021
TVAE	0.4472 ± 0.0220	-0.0659	<0.001	0.4704 ± 0.0300	-0.0168	0.246
HYBRID	0.4902 ± 0.0514	-0.0229	0.640	0.5252 ± 0.0222	+0.0380	0.017

Δ vs REAL: mean difference compared to the baseline trained with the same classifier. The p-values are corrected with Holm within each classifier; † REAL-CW: uncorrected p-value.

Comparison with the Cost-sensitive Baseline. REAL-CW Significantly Increases Recall in All Non-saturated Comparisons (e.g., DS1: from 0.1617 to 0.7938) without the substantial loss of AUC-PR observed with the synthetic methods in DS1 and DS4, and it obtains the highest F1-Score and G-mean in DS1 and DS4; in DS5 it is comparable to SMOTE-NC.

V. Discussion

Together, the results show that data synthesis does not work the same in all cases, but rather depends on the type of dataset, the level of imbalance, and how much information is actually available. In terms of fidelity, SMOTE-NC performs better when the data has a well-defined numerical part, and the imbalance is not as extreme as in DS1. There, it manages to maintain the structure of the data well with a low Wasserstein Distance, which basically shows that interpolation works well when the database is "stable". On the other hand, when the scenario becomes more complicated and there are many categorical variables, as in DS2, the CTGAN-based methods (HYBRID and CTGAN) reproduce the categorical proportions better, which is consistent with the conditional generator of CTGAN, designed to sample the categories of discrete columns in a balanced way [37].

When moving to predictive utility, the statistical validation of Section IV-D forces us to distinguish two planes that initially seemed to be one. In DS5, HYBRID and CTGAN improve the AUC-PR of XGBoost significantly (HYBRID: 0.5252 versus 0.4872) but not that of CatBoost, and they do not exceed the best baseline; the hypothesis that synthesis improves the global ordering under extreme scarcity is therefore only partially supported. The consistent effect is on sensitivity: CTGAN raises Recall in DS1 from 0.1617 to 0.5835, and SMOTE-NC in DS5 from 0.2409 to 0.6273. This suggests that the real contribution of the synthesis is to relocate the decision boundary to regions where the base model lacked information to risk a positive prediction.

This interpretation is reinforced by REAL-CW, which produces a similar or larger displacement without generating records, since oversampling and class reweighting act on the same mechanism: the relative weight of the minority class in the loss.

The dependence on the classifier in DS5 admits two explanations. First, the differences (0.023 to 0.038 in AUC-PR) are of the same order as the variability between runs, and the test partition contains only 44 positive records. Second, CatBoost builds symmetric trees, which limit the isolation of small regions, whereas XGBoost grows asymmetric trees that can exploit the dispersed CTGAN records in 95 dimensions; moreover, the XGBoost baseline is weaker in this dataset. This second explanation is a hypothesis, not a verified mechanism.

Regarding extreme imbalance, the generators learn from 22,622 minority records in DS1 but only 141 in DS5 (about 1.5 per feature) and 69 in DS2. This explains the large fidelity errors of the deep models in DS5 and DS2, and why SMOTE-NC, which only interpolates between existing neighbours, obtains the largest gain in Recall in DS5. The behavior in DS5 therefore reflects extreme imbalance, a very small minority class, and high dimensionality together, and, being the only dataset above 20:1, it should be confirmed in additional datasets.

In this sense, the hybrid SMOTE-NC-CTGAN approach is the most conservative synthetic method: it keeps precision close to the baseline and loses less AUC-PR than CTGAN and TVAE in DS1, but its gain in Recall is small, and its only significant improvement in AUC-PR appears in DS5 with XGBoost. SMOTE-NC is the most reliable option when the minority class is reasonably represented.

From a practical point of view, this displacement has a cost: in DS1, CTGAN lowers Precision from 0.56 to 0.33, so about two out of three positive predictions are false alarms, which may mean unnecessary clinical tests or the rejection of solvent clients. The choice of method should therefore consider the relative cost of both errors, and reweighting the classes should be considered first, since it obtains a similar displacement at a lower cost.

Of course, the study was not free of limitations. The datasets come from a curated repository (UCI) with few missing values and little noise, so the patterns may differ in messier real-world health and finance data [11]; only one dataset exceeds 20:1; the hyperparameter grid of the generators was small [9], [20]; and a single train/test partition and the default threshold were used. Finally, privacy was not addressed; future work should include multi-class problems and differential privacy.

VI. Conclusion

From the results obtained, several conclusions can be drawn about the impact of data synthesis on imbalanced classification problems: Dependence on the Ratio and Behavior under Extreme Imbalance: the Performance of Synthesis Techniques Depends Directly on the Level of Imbalance. SMOTE-NC Preserves the Structure of the Data Best in All Scenarios and Is the Most Reliable Option in Moderate Imbalance. When the Imbalance Is Extreme (29.99:1), CTGAN and the Hybrid Approach Significantly Improve the AUC-PR of XGBoost but Not that of CatBoost, and They Do Not Exceed the Best Baseline; This Behavior Is Therefore Classifier-dependent. This Scenario Is Represented by a Single Dataset with a Very Small Minority Class, Which Limits the Generality of This Conclusion.

Diversity Must Be Interpreted Together with Fidelity: in High-scarcity Scenarios, the GAN-based and Hybrid Models Generate the Most Diverse Records, but This Diversity Coincides with the Largest Fidelity Errors and Translates into a Significant Gain in AUC-PR Only in One Scenario (DS5 with XGBoost). Therefore, Diversity Alone Is Not an Indicator of the Usefulness of Synthetic Data.

Hybrid Approach Balance: the SMOTE-NC-CTGAN Model Is the Most Conservative of the Synthetic Methods: it Keeps Precision Close to that of the Baseline and achieves the Only Significant Improvements in AUC-PR, together with CTGAN, in the Most Critical Scenario with XGBoost (DS5: 0.5252 Versus 0.4872). However, Its Gain in Recall Is Small, and Its Advantage Depends on the Classifier, so Its Merit Lies in This Specific Scenario and Not in a General Superiority over the Baseline.

Improvement in the Detection of the Minority Class: Finally, and as a Main Result of This Research, it Is Concluded that Data Synthesis Improves the Model's Ability to identify the Minority Class in a Statistically Robust Way. Recall Increases Significantly in 27 of 40 Comparisons and Never Decreases, at the Cost of AUC-PR and of Precision (lower in 25); Reweighting the Classes Achieves a Comparable Displacement. This Trade-off Is Favorable Where the Cost of a False Negative Dominates, Such as in clinical diagnosis and Credit Risk, and must be weighed against the Cost of False Alarms.

REFERENCES

[1] M. G. Wijaya, M. F. Pinaringgi, A. Y. Zakiyyah, and Meiliana, "Comparative analysis of machine learning algorithms and data balancing techniques for credit card fraud detection," Procedia Computer Science, vol. 245, pp. 677–688, 2024, https://doi.org/10.1016/j.procs.2024.10.294

[2] H. Hairani, T. Widiyaningtyas, and D. D. Prasetya, "Addressing class imbalance of health data: a systematic literature review on modified synthetic minority oversampling technique strategies," JOIV: Int. J. Informat. Visualiz., vol. 8, no. 3, pp. 1310–1318, 2024, http://dx.doi.org/10.62527/joiv.8.3.2283

[3] A. G. Akintola, "Performance analysis of machine learning methods with class imbalance problem in Android malware detection," Int. J. Interact. Mob. Technol., vol. 16, no. 10, pp. 140–162, May 2022, https://doi.org/10.3991/ijim.v16i10.29687

[4] A. G. Udu, M. T. Salman, M. K. Ghalati, A. Lecchini-Visintini, D. R. Siddle, and H. Dong, "Emerging SMOTE and GAN variants for data augmentation in imbalance machine learning tasks: A review," IEEE Access, vol. 13, pp. 113838–113853, 2025, https://doi.org/10.1109/ACCESS.2025.3584532.

[5] N. Peppes, T. Alexakis, E. Daskalakis, and E. Adamopoulou, "Evaluating synthetic malicious network traffic generated by GAN and VAE models: a data quality perspective," Future Internet, vol. 17, no. 12, p. 561, 2025, https://doi.org/10.3390/fi17120561

[6] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey," J. Big Data, vol. 9, p. 98, 2022, https://doi.org/10.1186/s40537-022-00648-6

[7] T. Hasan and S. Tasnim, "Real-time explainable IoT security with machine learning and CTGAN-enhanced detection for resource-constrained devices," Ad Hoc Netw., vol. 178, p. 103937, 2025, https://doi.org/10.1016/j.adhoc.2025.103937

[8] D. Cullen, J. Halladay, N. Briner, R. Basnet, J. Bergen, and T. Doleck, "Evaluation of synthetic data generation techniques in the domain of anonymous traffic classification," IEEE Access, vol. 10, pp. 129612–129625, 2022, https://doi.org/10.1109/ACCESS.2022.3228507

[9] D.-H. Won, K.-S. Shin, and S. Youm, "Synthetic data augmentation for imbalanced tabular data: a comparative study of generation methods," Electronics, vol. 15, no. 4, p. 883, 2026, https://doi.org/10.3390/electronics15040883

[10] W. A. W. A. Bakar, N. L. N. Josdi, M. Man, and E. A. Kadir, "Evaluating CTGAN-generated synthetic data for heart disease prediction: fidelity, predictive utility, and feature preservation," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 12, 2025, http://dx.doi.org/10.14569/IJACSA.2025.0161296.

[11] M. Miletic and M. Sariyar, "Challenges of using synthetic data generation methods for tabular microdata," Appl. Sci., vol. 14, no. 14, p. 5975, 2024, https://doi.org/10.3390/app14145975

[12] A. Sharma, P. K. Singh, and R. Chandra, "SMOTified-GAN for class imbalanced pattern classification problems," IEEE Access, vol. 10, pp. 30655–30665, 2022, https://doi.org/10.1109/ACCESS.2022.3158977

[13] D.-K. Kim and Y. K. Chung, "A GAN-based hybrid approach for addressing class imbalance in machine learning," Comput. Sci. Inf. Syst., vol. 22, no. 3, pp. 673–691, 2025, https://doi.org/10.2298/CSIS231113014K

[14] I. B. Mustapha, S. Hasan, H. S. Y. Nabbus, M. M. A. Montaser, S. O. Olatunji, and S. M. Shamsuddin, "Investigating group distributionally robust optimization for deep imbalanced learning: a case study of binary tabular data classification," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 2, 2023, https://doi.org/10.14569/ijacsa.2023.0140286

[15] H. A. Eren, H. İ. Emek, and S. B. Keser, "Comparison of synthetic data generation techniques for obesity level prediction based on dietary habits and physical status," Artif. Intell. Health, vol. 2, no. 4, pp. 47–74, 2025, https://doi.org/10.36922/AIH025140027

[16] S. B. Belhaouari, A. Islam, K. Kassoul, A. Al-Fuqaha, A. Bouzerdoum, "Oversampling techniques for imbalanced data in regression," Expert Syst. Appl., vol. 252, part B, p. 124118, 2024, https://doi.org/10.1016/j.eswa.2024.124118

[17] K. Omari, C. Taoussi, and A. Oukhatar, "Comparative analysis of undersampling, oversampling, and SMOTE techniques for addressing class imbalance in phishing website detection," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 2, 2025, http://dx.doi.org/10.14569/IJACSA.2025.0160276

[18] D. Utari, "Integration of SVM and SMOTE-NC for classification of heart failure patients," Barekeng: J. Ilmu Mat. Ter., vol. 17, no. 4, pp. 2263–2272, 2023, https://doi.org/10.30598/barekengvol17iss4pp2263-2272

[19] H. Abtahi, M. Edalatifard, M. Gholamzadeh, and E. Khakvatan, "Predicting zygosity based on physical similarity of twin pairs with the aid of machine learning methods," BMC Med. Inform. Decis. Mak., vol. 26, no. 1, p. 32, 2025, https://doi.org/10.1186/s12911-025-03334-8

[20] P. Yadav, M. Gaur, R. K. Madhukar, G. Verma, P. Kumar, N. Fatima, S. Sarwar, and Y. R. Dwivedi, "Rigorous experimental analysis of tabular data generated using TVAE and CTGAN," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 4, 2024, http://dx.doi.org/10.14569/IJACSA.2024.01504125.

[21] M. Alqulaity and P. Yang, "Enhanced conditional GAN for high-quality synthetic tabular data generation in mobile-based cardiovascular healthcare," Sensors, vol. 24, no. 23, p. 7673, 2024, https://doi.org/10.3390/s24237673

[22] P. C. Y. Cheah, Y. Yang, and B. G. Lee, "Enhancing financial fraud detection through addressing class imbalance using hybrid SMOTE-GAN techniques," Int. J. Financial Stud., vol. 11, no. 3, p. 110, 2023, https://doi.org/10.3390/ijfs11030110

[23] G. Eom and H. Byeon, "Searching for optimal oversampling to process imbalanced data: generative adversarial networks and synthetic minority over-sampling technique," Mathematics, vol. 11, no. 16, p. 3605, 2023, https://doi.org/10.3390/math11163605

[24] H.-Y. Cho and Y.-H. Kim, "A genetic algorithm to optimize SMOTE and GAN ratios in class imbalanced datasets," in Proc. 2020 Genet. Evol. Comput. Conf. Companion (GECCO '20), 2020, pp. 33–34, https://doi.org/10.1145/3377929.3398153

[25] N. Uddin, P. Mahmud, M. Ahammed, et al., "Explainable ensemble learning model for cardiovascular disease prediction with feature optimization and data balancing," Discov. Comput., vol. 29, p. 44, 2026, https://doi.org/10.1007/s10791-026-09930-0

[26] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, "Comparative analysis using various performance metrics in imbalanced data for multi-class text classification," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 6, 2023, http://dx.doi.org/10.14569/IJACSA.2023.01406116

[27] I. Al-Hurani, S. H. ElFar, A. Alkhateeb, and S. Ikki, "AE-CTGAN: autoencoder–conditional tabular GAN for multi-omics imbalanced class handling and cancer outcome prediction," Algorithms, vol. 19, no. 2, p. 95, 2026, https://doi.org/10.3390/a19020095

[28] Y. Lin, C. Huang, H. Tian, et al., "Improving generalization of polyp detection via conditional StyleGAN augmented training," npj Digit. Med., vol. 9, p. 113, 2026, https://doi.org/10.1038/s41746-025-02293-8

[29] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu.

[30] Centers for Disease Control and Prevention (CDC), "Behavioral Risk Factor Surveillance System (BRFSS) 2014 Data and Documentation," 2014. [Online]. Available: https://www.cdc.gov/brfss/annual_data/annual_2014.html.

[31] "CDC Diabetes Health Indicators," UCI Machine Learning Repository, 2017. [Online]. Available: https://doi.org/10.24432/C53919.

[32] S. Borzooei and A. Tarokhian, "Differentiated Thyroid Cancer Recurrence," UCI Machine Learning Repository, 2023. [Online]. Available: https://doi.org/10.24432/C5632J.

[33] National Institute of Allergy and Infectious Diseases (NIAID), "A Randomized, Double-Blind Phase II/III Trial of Monotherapy vs. Combination Therapy With Nucleoside Analogs in HIV-Infected Persons With CD4 Cells of 200–500/mm³," ClinicalTrials.gov, NCT00000625, 2021.

[34] S. Hammer et al., "AIDS Clinical Trials Group Study 175," UCI Machine Learning Repository, 2023. [Online]. Available: https://doi.org/10.24432/C5ZG8F.

[35] I. Yeh, "Default of Credit Card Clients," UCI Machine Learning Repository, 2009. [Online]. Available: https://doi.org/10.24432/C55S3H.

[36] J. J. S. Huang, "Taiwanese Bankruptcy Prediction," UCI Machine Learning Repository, 2020. [Online]. Available: https://doi.org/10.24432/C5004D.

[37] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in Advances in Neural Information Processing Systems, vol. 32, 2019.

[38] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLoS ONE, vol. 10, no. 3, p. e0118432, 2015, https://doi.org/10.1371/journal.pone.0118432

[39] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in Proc. 14th Int. Conf. Machine Learning (ICML), 1997, pp. 179–186.

APPENDIX

TABLE VI. HYPERPARAMETERS SELECTED FOR EACH DATASET (GRID SEARCH, LOGISTIC REGRESSION PROXY, AUC-PR ON VALIDATION)

Dataset	Minority records (train-in)	SMOTE-NC k	CTGAN (epochs, batch, pac)	TVAE (epochs, batch, emb. dim.)	HYBRID CTGAN stage (epochs, batch, pac)	α (HYBRID)
DS1	22,622	3	300, 500, 10	500, 256, 128	500, 1000, 10	0.5
DS2	69	7	500, 1000, 10	500, 512, 256	300, 500, 10	0.5
DS3	333	7	500, 500, 10	300, 256, 128	500, 1000, 10	0.5
DS4	4,247	3	300, 500, 10	500, 256, 128	500, 1000, 10	0.5
DS5	141	3	300, 500, 10	500, 512, 256	300, 500, 10	0.5

TABLE VII. PREDICTIVE PERFORMANCE ON THE TEST PARTITION WITH XGBOOST (MEAN ± SD OVER 10 RUNS)

Dataset	Ratio	Method	AUC-PR	Recall	Precision	F1-Score	Specificity	G-mean	Bal. Acc.
DS1	6.18	REAL	0.4219 ± 0.0005	0.1670 ± 0.0015	0.5518 ± 0.0031	0.2564 ± 0.0021	0.9780 ± 0.0002	0.4041 ± 0.0019	0.5725 ± 0.0008
		REAL-CW	0.4228 ± 0.0006	0.7852 ± 0.0009	0.3104 ± 0.0005	0.4449 ± 0.0006	0.7176 ± 0.0007	0.7507 ± 0.0005	0.7514 ± 0.0005
		SMOTE-NC	0.4117 ± 0.0009	0.3372 ± 0.0029	0.4599 ± 0.0029	0.3891 ± 0.0015	0.9359 ± 0.0012	0.5618 ± 0.0021	0.6365 ± 0.0010
		CTGAN	0.3372 ± 0.0073	0.5591 ± 0.0236	0.3247 ± 0.0086	0.4105 ± 0.0060	0.8114 ± 0.0139	0.6733 ± 0.0093	0.6853 ± 0.0063
		TVAE	0.3189 ± 0.0082	0.4341 ± 0.0403	0.3263 ± 0.0092	0.3715 ± 0.0127	0.8545 ± 0.0173	0.6082 ± 0.0231	0.6443 ± 0.0119
		HYBRID	0.4143 ± 0.0022	0.1788 ± 0.0040	0.5297 ± 0.0061	0.2673 ± 0.0042	0.9743 ± 0.0010	0.4174 ± 0.0045	0.5766 ± 0.0016
DS2	2.55	REAL	0.9939 ± 0.0008	0.9545 ± 0.0000	0.9818 ± 0.0235	0.9679 ± 0.0115	0.9927 ± 0.0094	0.9734 ± 0.0046	0.9736 ± 0.0047
		REAL-CW	0.9929 ± 0.0011	0.9545 ± 0.0000	0.9504 ± 0.0131	0.9524 ± 0.0067	0.9800 ± 0.0057	0.9672 ± 0.0028	0.9673 ± 0.0029
		SMOTE-NC	0.9937 ± 0.0016	0.9545 ± 0.0000	0.9682 ± 0.0220	0.9612 ± 0.0107	0.9873 ± 0.0088	0.9708 ± 0.0043	0.9709 ± 0.0044
		CTGAN	0.9934 ± 0.0023	0.9545 ± 0.0000	0.9553 ± 0.0290	0.9547 ± 0.0145	0.9818 ± 0.0121	0.9681 ± 0.0060	0.9682 ± 0.0061
		TVAE	0.9930 ± 0.0012	0.9545 ± 0.0000	0.9773 ± 0.0240	0.9656 ± 0.0117	0.9909 ± 0.0096	0.9725 ± 0.0047	0.9727 ± 0.0048
		HYBRID	0.9942 ± 0.0032	0.9500 ± 0.0144	0.9686 ± 0.0298	0.9589 ± 0.0139	0.9873 ± 0.0123	0.9684 ± 0.0077	0.9686 ± 0.0076
DS3	3.10	REAL	0.8637 ± 0.0034	0.7337 ± 0.0129	0.8075 ± 0.0066	0.7687 ± 0.0070	0.9438 ± 0.0028	0.8321 ± 0.0067	0.8387 ± 0.0057
		REAL-CW	0.8607 ± 0.0026	0.7625 ± 0.0129	0.7806 ± 0.0093	0.7714 ± 0.0081	0.9312 ± 0.0039	0.8426 ± 0.0068	0.8468 ± 0.0062
		SMOTE-NC	0.8612 ± 0.0034	0.7587 ± 0.0071	0.7797 ± 0.0088	0.7690 ± 0.0066	0.9312 ± 0.0033	0.8405 ± 0.0045	0.8449 ± 0.0042
		CTGAN	0.8549 ± 0.0118	0.7606 ± 0.0132	0.7552 ± 0.0206	0.7577 ± 0.0131	0.9207 ± 0.0086	0.8368 ± 0.0083	0.8406 ± 0.0079
		TVAE	0.8523 ± 0.0055	0.7567 ± 0.0144	0.7710 ± 0.0152	0.7637 ± 0.0125	0.9278 ± 0.0059	0.8379 ± 0.0089	0.8423 ± 0.0084
		HYBRID	0.8588 ± 0.0047	0.7529 ± 0.0111	0.7726 ± 0.0167	0.7625 ± 0.0087	0.9287 ± 0.0072	0.8362 ± 0.0058	0.8408 ± 0.0053
DS4	3.52	REAL	0.5498 ± 0.0021	0.3656 ± 0.0037	0.6547 ± 0.0053	0.4692 ± 0.0042	0.9453 ± 0.0009	0.5878 ± 0.0031	0.6554 ± 0.0021

		REAL-CW	0.5479 ± 0.0023	0.5910 ± 0.0024	0.4812 ± 0.0042	0.5305 ± 0.0029	0.8190 ± 0.0030	0.6958 ± 0.0018	0.7050 ± 0.0019
		SMOTE-NC	0.5286 ± 0.0026	0.4695 ± 0.0041	0.5288 ± 0.0077	0.4974 ± 0.0048	0.8812 ± 0.0034	0.6432 ± 0.0032	0.6753 ± 0.0028
		CTGAN	0.5371 ± 0.0048	0.3826 ± 0.0046	0.6339 ± 0.0055	0.4772 ± 0.0048	0.9373 ± 0.0011	0.5988 ± 0.0037	0.6599 ± 0.0025
		TVAE	0.5234 ± 0.0033	0.4099 ± 0.0072	0.5862 ± 0.0031	0.4824 ± 0.0050	0.9178 ± 0.0019	0.6134 ± 0.0049	0.6639 ± 0.0029
		HYBRID	0.5339 ± 0.0059	0.3786 ± 0.0063	0.6297 ± 0.0099	0.4728 ± 0.0053	0.9367 ± 0.0030	0.5955 ± 0.0046	0.6577 ± 0.0028
DS5	29.99	REAL	0.4872 ± 0.0173	0.3091 ± 0.0389	0.6388 ± 0.0489	0.4158 ± 0.0420	0.9942 ± 0.0010	0.5534 ± 0.0341	0.6516 ± 0.0195
		REAL-CW	0.4662 ± 0.0176	0.5000 ± 0.0321	0.5046 ± 0.0216	0.5020 ± 0.0246	0.9836 ± 0.0011	0.7010 ± 0.0225	0.7418 ± 0.0160
		SMOTE-NC	0.4768 ± 0.0101	0.6000 ± 0.0287	0.4800 ± 0.0113	0.5331 ± 0.0160	0.9783 ± 0.0010	0.7660 ± 0.0182	0.7892 ± 0.0141
		CTGAN	0.5137 ± 0.0162	0.2682 ± 0.0352	0.6123 ± 0.0618	0.3722 ± 0.0420	0.9943 ± 0.0011	0.5154 ± 0.0345	0.6312 ± 0.0177
		TVAE	0.4704 ± 0.0300	0.3705 ± 0.0340	0.6326 ± 0.0555	0.4661 ± 0.0342	0.9927 ± 0.0017	0.6059 ± 0.0276	0.6816 ± 0.0169
		HYBRID	0.5252 ± 0.0222	0.3091 ± 0.0287	0.6448 ± 0.0484	0.4167 ± 0.0275	0.9942 ± 0.0013	0.5538 ± 0.0254	0.6517 ± 0.0141

Same protocol and notation as Table III.

A Deep Learning-Based Real-Time Surveillance Framework for Mitigating Herbivorous Animal Intrusions in Rice Nursery Fields

Muhammad Juman Jhatial, Riaz Ahmed Shaikh, Rafaqat Hussain Arain
Institute of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan

***Abstract*—The rice nursery stage is a very critical stage of the rice growing cycle, as intrusion of herbivorous animals can lead to substantial economic losses and yield losses. Manual guarding and physical fencing are inefficient and ineffective. Real-Time Animal Intrusion Detection Network (RAID-Net) is an end-to-end intelligent agricultural surveillance system that consists of an integration of YOLO and a multi-task CNN to detect and recognize herbivorous animal intrusion in open-field rice nursery fields in real time. To this end, a novel and perfectly balanced dataset of 14,000 images with five different herbivore species (Buffalo, Camel, Cow, Goat, and Sheep) was curated, and a variety of seven YOLO models and five CNN architectures were benchmarked to identify the optimal detection and classification backbones. The model design includes a lightweight YOLOv8 Nano object detector and a custom-designed Multi-task Convolutional Neural Network (CNN) for species classification and behavior detection grazing and moving. The results of the Customized Multi-task CNN on cropped animal regions showed 97.72% species classification accuracy and 98.14% behaviour recognition accuracy. The RAID-Net with YOLO detection-aware alert pipeline yielded 90.05% species accuracy, 94.31% behaviour accuracy, and 82.73% hybrid accuracy, with an end-to-end latency of 17.61 ms and an inference rate of 56.79 FPS on edge-compatible hardware with an alert precision of 98% and recall of 89%. The system differentiates between high-threat grazing and low-threat moving behaviors, automatically activating a siren deterrent and sending instant WhatsApp/SMS alerts to farmers, providing an automated, scalable, and very accurate solution for precision agriculture and crop protection.**

Keywords*—Precision agriculture; deep learning; Yolo model; convolutional neural networks

I. Introduction

Rice farming is the basic livelihood activity for food security in Pakistan and plays a significant role in rural economies. The nursery stage is crucial for the establishment of seedlings, with a duration of 20-30 days.

At this time, rice plants are delicate and prone to grazing and trampling by domestic herbivorous animals [1]. Out of many herbivorous animals, five such animals were chosen for this study as they are seen with regularity in rice nursery fields and have a marked effect on the damage to the crop (Buffalo, Camel, Cow, goat, and Sheep). Illustrated in Fig. 1.



Fig. 1. Dataset images.

The rice nursery stage is one of the most critical periods of the rice production cycle, especially in open field conditions where the intrusions of herbivorous animals may lead to damage to young seedlings, which may be irreversible and fast. Domestic animals like buffalo, camel, cow, goat, and sheep often intrude into the nursery fields for food and have strong grazing ability, which easily strips newly planted rice plants, resulting in huge economic losses for farmers in many rural agricultural areas. Current protection measures are often traditional, reactive, and labour-intensive, such as manual guarding, fencing, and scare devices [2]. They are labour-intensive, time-consuming, and difficult to scale up to large areas of agriculture, particularly when animals can be in fields without being monitored and cannot be seen at night or in unsuitable weather. Recently, computer vision and smart agriculture technologies have emerged with potential, but most of the proposed methods are either object detection or species classification and fail to consider behavior analysis, severity assessment of intrusions, and automatic alert generation. Hence, there is a critical need for a specialized end-to-end intelligent surveillance framework that can detect, identify, and assess threats based on species behavior for proactively protecting the rice nursery field in real-time and enhancing agricultural resilience.

A. Computer Vision and Deep Learning

The past few years have seen tremendous progress in computer vision and deep learning, which has led to the development of robust object detection and image classification algorithms that can perform these tasks in real time with high accuracy in challenging scenarios. In agriculture, intelligent systems can also be used to replace manual field supervision using the detection and tracking of animals like cows, goats, and buffalo in rice paddy fields, as shown in Fig. 2, which is a promising solution for agricultural monitoring.



Fig. 2. Computer vision system monitoring a rice paddy field.

However, the existing agricultural monitoring frameworks are still limited in their functions. Many systems are limited to the mere detection of animal movement, and not necessarily on the basis of what it means in terms of animal behavior, which makes it challenging to distinguish harmless animal movement from destructive grazing behavior. On the other hand, there are also studies that focus on species classification but do not have the capability of spatial localization because they work on images that have been pre-segmented or cropped instead of identifying the animals in large-scale real scenes in the agricultural field. Consequently, existing methods focus on detection or classification only, and there is a significant research gap in how to combine spatial localization with species identification and behaviour understanding into a single entity. Considering this shortcoming, this study presents a comprehensive deep learning-based surveillance system, especially for protecting the rice nursery field. An annotated herbivore intrusion dataset of five domestic animals was built to facilitate training and evaluation of the model. Various detection models and CNN architectures based on YOLO were extensively benchmarked to choose the best configurations considering the accuracy, inference speed, and computational efficiency. Based on the above results, a new hybrid model fused YOLO for detection and a multi-task CNN for classification, which can simultaneously localize, identify the species, and recognize the behaviors of the animals, was constructed. Moreover, a behavior-aware threat assessment mechanism was added to distinguish the difference in movement between the animal moving and the destructive use of the animal to graze, enabling context-aware decision-making. Last but not least, a real-time alert and deterrence pipeline was put in place, comprising automated siren activation and mobile notifications to enable farmers to respond immediately, rather than relying on passive observation, leading to a shift from passive monitoring to proactive crop protection.

II. Related Work

The integration of artificial intelligence (AI), machine learning (ML), and deep learning (DL) in agriculture has emerged as an effective solution to challenges related to population growth, climate change, and resource scarcity, improving crop management, resource optimization, and pest control. In this comprehensive review, the authors investigate the potential of machine learning and deep learning applications for agricultural use. Food security is still defined largely by crops, livestock, and fisheries, but these are subjected to serious challenges including climate variability, scarcity of resources, and sustainability issues. Recent research has shown that AI can be used to solve problems, from simple machine learning to more complex deep learning and vision-language models, in various fields like crop disease detection, livestock health monitoring, and aquatic species management. Yet, challenges such as data variability, scalability constraints, and deployment constraints underscore the importance of multimodal data integration and flexible AI solutions for a variety of farming contexts.

A. AI in Agriculture: A Survey of Deep Learning Techniques for Crops, Fisheries and Livestock.

Consequently, deep learning has emerged as a significant technology in precision agriculture, especially when it comes to the early detection of crop diseases that pose threats to yield and quality. This study used multi-modal approaches on potato, tomato, grape, apple, and peach leaves and found that VGG16, MobileNetV2, Xception, and ResNet all performed quite well, with the metrics of Accuracy, Precision, Recall, and F1-Score. The findings indicate that VGG16 achieved 99% efficiency, suggesting that scalable, non-invasive AI systems could provide valuable plant disease management tools, all supported by farmer trust in AI systems.

Judging from the title, the authors have a very creative methodology for enhancing agricultural disease detection. Judging from the title, the authors have a very creative methodology for enhancing agricultural disease detection.

Food sustainability is a growing global challenge due to rapid urbanization and globalization, and AI systems can provide new solutions to improve productivity and food security. Crop Diseases like chlorosis continue to pose a significant threat in India, where more than half of the population relies on agriculture for their livelihood, despite significant strides in the past, including the Green Revolution. The existing detection method has shortcomings in accuracy, and deep learning technology is a more reliable and scalable solution for disease detection in the context of variable environments and climates.

B. Leveraging Deep Learning and Neural Networks for Real-Time Crop Monitoring in Smart Agricultural Systems Srinivas Kalyan Yellanki

Precision agriculture and plant disease diagnosis rely heavily on early detection and accurate diagnosis, especially when plant diseases are a significant threat to food security worldwide. In recent years, studies have emerged that show the ability of computer vision and deep learning methods such as CNNs, transformer vision, and GANs combined with imaging technologies like RGB, multispectral, and hyperspectral sensors. A great deal of research reviews have documented progress and identified challenges associated with the use of these models and have offered useful insights on how these models are applicable for sustainable agricultural production. The use of deep learning and computer vision has been proposed as a comprehensive review of the techniques, models, and trends in precision agriculture for the purposes of plant disease detection. Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models and trends in precision agriculture. Artificial Intelligence (AI) and Deep Learning (DL) are two technologies that have the promise of revolutionizing agriculture and offering new solutions to some of its biggest problems, including productivity, sustainability, labour shortages, and environmental monitoring. Intelligent systems, such as computer vision, machine learning, and edge computing, have been developed to perform various tasks like crop classification, weed and pest detection, yield estimation, soil health analysis, and automatic irrigation control, etc., in recent years [1, 2]. Although these all yield different results, deep learning, particularly using Convolutional Neural Networks (CNNs), has been very effective for image processing applications such as plant disease recognition, fruit grading, phenol typing, livestock monitoring, and more [3] [4]. For object detection in real time, there are some popular frameworks like YOLO (You Only Look Once), Faster R-CNN, and SSD (Single Shot Multi-Box Detector) which are used for agriculture-related objects like plant, pest, fruit, and animal object detection [5]. The models can be used in precision agriculture: automatic identification and localization of certain entities in images or real video. Furthermore, AI-driven Unmanned Aerial Vehicles (UAVs), autonomous tractors, and robotic harvesters have started to change the dynamics of field operations, making them more data-informed and independent in how they are conducted. Moreover, Recurrent Neural Networks (RNNs) and LSTM models are applied to time series prediction tasks such as prediction of the weather, crop yield, irrigation scheduling, etc. [6]. But there remains a whole host of issues to be resolved, including: availability of large annotated data sets; regional and seasonal generalizability of models; ability to deploy models on low-cost edge devices suitable to rural environments. While there have been significant advances in crop and disease detection, the ability to detect livestock and/or animal incursions in some high-risk sites, e.g., rice nurseries, has not been well explored. This points to the need to develop more specialized and domain-specific deep learning systems to address the technical and practical issues in the agri-environment. 2.2 Animal detection or monitoring systems in place. The traditional methods of animal counting are still challenging in livestock management, and drone-based methods are also not easy. This study presents the Mask YOLOv7, an advanced object detection and instance segmentation model, for detecting and counting cattle in both controlled and uncontrolled settings, such as feedlots and open ranges. The experiment results reveal that the model has high accuracy of 93% and 95%, respectively, which makes it possible to apply it in livestock inventories and monitoring and also in intelligent farm management applications. Drone Vision System for Automated Cattle Detection and Counting using Mask YOLOv7 is an innovative technology that aims to monitor cattle from a distance using an unmanned aerial vehicle (UAV). Despite the importance of cows in daily life and nutrition, especially in India, maintaining healthy cows can be a big problem, and lameness is one of the most costly problems in dairy herds. This study introduces a Smart Cow Health Monitoring System (ScHeMoS) that uses an IoT sensor system and, in particular, the LSTM deep learning model to analyze the behavior and physiological parameters of the cow for the early detection of lameness. The model showed good accuracy of 92.45% and an F1 score of 0.92 on the self-collected set, suggesting its potential feasibility for practical farm management and animal welfare practices.

C. Schemos Smart Cow Health Monitoring System: An IoT-Based Cow Hoof Detection And Healthcare Alert System Using LSTM Network

Regularly checking on a pet is very important to lower health risks, but checking on external behavior can be misleading. In this study, a video-based monitoring system that extracts behavioral features from the image, detects anomalies, and alerts to deviations from common behavior is proposed using Mask R-CNN. Hamster experiments showed the effectiveness of the system in recognizing abnormal and normal movements, and future work plans involve increasing the sensitivity of the system to move towards recognizing more subtle movements. Monitoring systems for animals based on behavioral information and internal state information. Regularly checking on pets is important to minimize health risks, but often external behavior is a poor indicator of internal condition. In this study, a video-based monitoring system with Mask R-CNN was proposed to obtain behavioral features and detect anomalies, making alerts whenever there are anomalies in the behaviors. Experimental tests with hamsters showed the efficacy of the system to discriminate between "normal" and "abnormal" behaviors, and further developments would involve increasing the sensitivity of the system to subtle movements.

D. Research on Behavior Recognition and Online Monitoring System for Liaoning Cashmere Goats Based on Deep Learning

Automated cameras (e.g., camera traps) are also deployed for wildlife monitoring, but there are some limitations in their effectiveness, including imperfect detection, limited storage, and power supply. This work introduces a customizable camera system with YOLOv5 integrated into a Raspberry Pi, which is used for continuous monitoring of fast-moving organisms in field conditions and is demonstrated in mason bees. The system was proven to be reliable, with an average detection precision of 78%, and it was found to be a low-cost, modular, and researcher-friendly solution to commercial camera traps. Monitoring high-speed animals, creating a specific camera system, and developing an evaluation toolset.

In the past few years, the detection and monitoring of animals has become a rising concern, especially in the fields of livestock management, wildlife conservation, and agricultural protection. There are a large number of works that used deep learning methods, including convolutional neural networks (CNNs) and object detection methods like YOLO (You Only Look Once), SSD (Single Shot Detector), and Faster R-CNN to detect animals in images or video streams [1],[2]. These are usually used to monitor open pasture areas [3], forest areas [4], or urban wildlife zones [5]. Although they have shown promising results in detecting animals like cattle, deer, or wild boars, most of the existing models are developed for generalized detection tasks, e.g., animal/non-animal or livestock/wildlife.

Moreover, the current animal monitoring systems are typically designed for controlled conditions, high-quality video resolution, and background stability, which are seldom observed in the real agricultural environment [6]. Many models are unable to be optimized for inference speed on edge devices, making it difficult to obtain a good accuracy rate. There is a lack of lightweight architectures and applications that are mobile-friendly and suitable for real-time application, especially in rural rice fields [7].

Furthermore, the problem of species-level identification of herbivorous animals (buffalo, cow, goat, sheep, and camel), which are visually similar in some aspects, is significant for prioritizing alerts based on the severity of crop damage, and few studies address this issue. In the same way, in the majority of works, behavioral analysis, including animal movement tracking or prediction of grazing, is still not well studied. Some work has used GPS collars or radio frequency identification (RFID) to track livestock [8], but such methods are not scalable and not suitable for free-roaming livestock entering rice fields. There is also a lack of real-time farm monitoring solutions, including the use of early warnings via SMS or mobile app, which further diminishes the usefulness of these solutions in practice [9].

III. Materials and Methods

A real-time deep learning-based surveillance framework, RAID-Net, is proposed to protect rice nursery fields from domestic herbivorous animal intrusions like buffalo, camels, cows, goats, and sheep. The architecture is built on a multi-stage pipeline architecture with integration of object detection, behavioral analysis, and automated response mechanisms, as shown in Fig. 3.

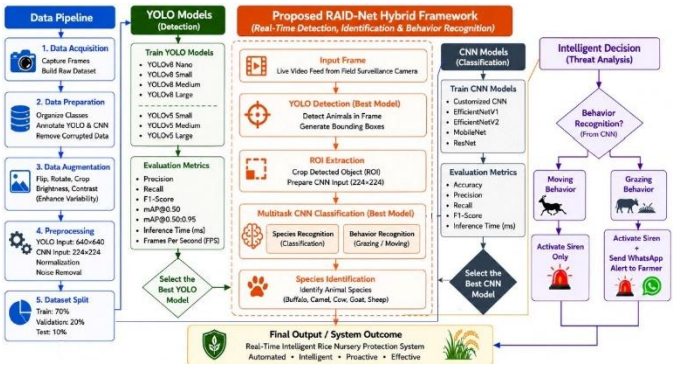


Fig. 3. Flowchart of research methodology and development pipeline.

This figure shows the five major steps in the research lifecycle: (Phase 1), Data Pipeline (acquisition and augmentation), YOLO testing and benchmarking (Phase 2), and CNN testing and benchmarking (Phase 3). It also explains the incorporation of these models in the RAID-Net (Phase 4). The final logic-based alert generation system (Phase 5) for protection of rice nursery fields.

A. Phase 1: Data Pipeline

The data pipeline can be used to systematically move images from acquisition to preprocessing to model inference, which makes it easy to detect and identify herbivorous animals (Fig. 4).

B. Data Collection

The data set contains a total of 14000 RGB images from various sources, such as rural field areas, open grazing areas, and publicly available data sets [14, 15, 16, 17]. The images are classified into 5 classes: Buffalo, Camel, Cow, Goat, and Sheep, and each class contains approximately 2800 images, which is a nice balance for training and testing of the model. Shown in Tables I and II.

TABLE I. Dataset Acquisition Characteristics

Source	Geographic location	Acquisition	Device source	Day/Night	Images
Rural field dataset	Larkana, Sindh, Pakistan	Field	Digital camera	Day	3,420
Open grazing dataset	Dadu, Sindh, Pakistan	Field	Smartphone	Day	3,680
Public Dataset A	Kaggle.com	Public dataset	Internet	Variable	3,510
Public Dataset B	Shutterstock.com	Public dataset	Internet	Variable	3,390
Total					**14000**



Fig. 4. Raw images.

TABLE II. CLASS-WISE DISTRIBUTION OF DATASET

Animal Class	Number of Images
Buffalo	2800
Camel	2700
Cow	2900
Goat	2700
Sheep	2800
Total	14000

C. Dataset Annotation and Labelling

The full set of data is systematically annotated and labelled for object detection using YOLO as well as CNN for classification. In the case of the YOLO models, the instances of animals in the whole dataset were manually cropped, and the class labels were manually assigned to the cropped objects. By having this annotation and labelling procedure, the data preparation and class representation for both the detection and the classification phases of the proposed framework were consistent [10], [11]. The dataset labelling is illustrated in Fig. 5 for classification models.



Fig. 5. Dataset labelling.

D. Dataset Splitting for Model Development.

To support proper model development, we divided the dataset as follows: 70% for training, 20% for validation, and 10% for testing [12], [13]. This is illustrated in Fig. 6 and Table III.



Fig. 6. Dataset splitting.

TABLE III. DATASET DISTRIBUTION FOR TRAINING, VALIDATION AND TESTING

Dataset Split	Percentage	Number of Images
Training set	70%	9800
Validation set	20%	2800
Testing set	10%	1400
Total		14000

E. Images Preprocessing and Augmentation

Pre-processing of images was done to prepare them for model training. This involved resizing the images to 640X640 for the YOLO model and 240X240 for the CNN model, and then normalizing the images. To improve the model's robustness and generalization, as shown in Fig. 7, data augmentation methods like horizontal flipping, brightness variations, and random cropping techniques were employed. Horizontal flipping, 15% brightness, and resizing are preprocessing steps [18],[19],[20].



Fig. 7. Data preprocessing and data augmentation.

F. Phase 2: Detection Module (YOLO)

The Detection Module in the proposed RAID-Net framework is based on the YOLO architecture [21]; Fig. 8 shows the YOLO Architecture below.

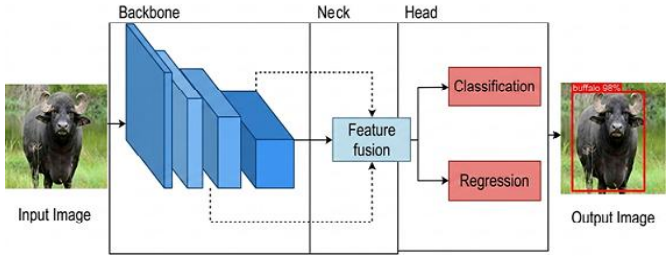


Fig. 8. YOLO architecture.

The system starts with a high-resolution camera that takes images from the field, which are then resized to 640×640 pixels and then represented as:

$$I \in R^{640\times640\times3}$$

The YOLOv8-Nano detector works on the input image and outputs a list of bounding boxes, confidences, and class probabilities:

$$f_{YOLO}(I) = \{B_i, C_i, S_i\}_{i=1}^{N}$$

where, B_i denotes the bounding box coordinates, C_i represents the confidence score, and S_i indicates the class probability for the ith detected object.

The composite loss function is used for the YOLO model:

$$L_{YOLO} = L_{BOX} + L_{OBJECT} + L_{CLS}$$

where, L_{BOX} represents bounding box regression loss, L_{OBJECT} denotes objectness loss, and L_{CLS} corresponds to classification loss. Together, they optimize the model to get accurate localization and reliable detection. Real-time agricultural surveillance of herbivorous animals that can be detected and effectively classified.

G. Phase 3: Multi-Task CNN Module (Species and Behavior Classification)

The Multi-Task CNN Classification Module is shown in Fig. 9 of the CNN model architecture [22].

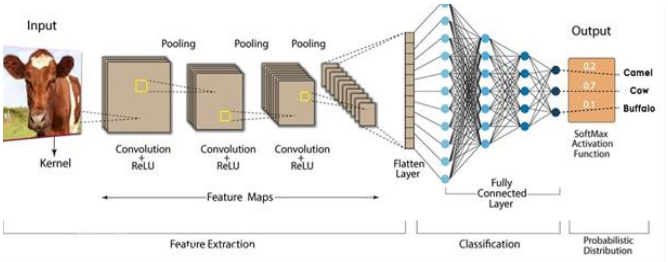


Fig. 9. CNN Architecture.

CNN-based classification stage for the RAID-Net framework:

$$F = CNN(ROI), \quad P_s = Softmax(W_s F + b_s), P_b = Softmax(W_b F + b_b)$$

$$L_{total} = \lambda 1 L_{species} + \lambda 2 L_{behavior}$$

F = feature vector from the cropped ROI (240 × 240 pixels)

P_s, P_b = predicted probabilities for species and behavior

W_s, b_s, W_b, b_b = classifier parameters

L_{total} = weighted multi − task loss balancing species and behavior learning

H. RAID-Net and Alert Phases

$$\text{Alert} = \begin{cases} Higt - Priority\ if\ S_{SPECIES} = Animal, S_{Behaviour} = Grazing, C_{Hybrid} \geq T_{Threshold} \\ Meduim - Priority\ if\ S_{SPECIES} = Animal, S_{Behaviour} = Moving, C_{Hybrid} \geq T_{Threshold} \end{cases}$$

$$\text{Response} = \begin{cases} Grazing: Acitvate\ Siren + Send\ WhatsApp\ Alert \\ Moving\ Activate\ Siren \end{cases}$$

$S_{species}$, $S_{behavior}$ = predicted species and behavior

C_{hybrid} = hybrid confidence score from fusion layer

$T_{threshold}$ = minimum confidence to trigger an alert

The outputs are connected to the real-world system, such as the siren hardware and WhatsApp API, allowing real-time deterrence and remote farmer notification. This is a combination of AI and rules, providing both reliability in operations and readability in agricultural field deployments. The RAID-Net architecture is shown in Fig. 10.

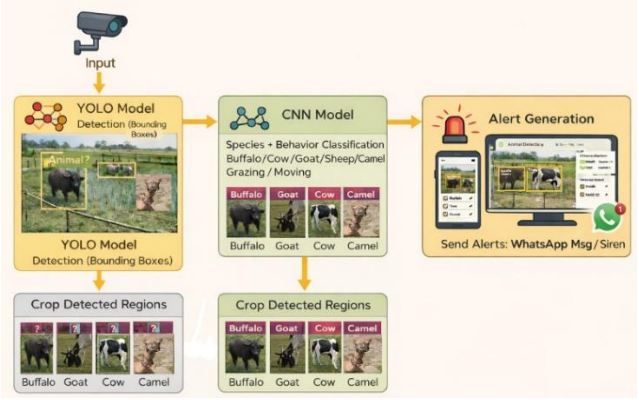


Fig. 10. Architectural framework of RAID-Net.

The diagram shows the actual operation of the surveillance system from the acquisition of the video stream from the input camera. It illustrates the multi-step process of passing the output of a YOLO model for object localization through a CNN model for fine-grained species and behavior classification, and then to either automated WhatsApp alerts or audible sirens, depending on the intrusion detected.

IV. Experimental Setup

To ensure the flexibility and scalability of the experimental setting, the proposed RAID-Net framework was implemented on cloud-based GPU platforms as well as a local computational system to facilitate reproducibility of the experiment. The implementation of YOLO based detection models and integration experiments in a hybrid setup consisted of using Google Colab with Python 3 and NVIDIA Tesla T4 GPU acceleration, which proved to be the most effective for training the YOLOv5 and YOLOv8 architectures with PyTorch and the Ultralytics framework. The multi-task CNN classification experiments were conducted on Kaggle Notebooks using GPU-enabled environments (P100/T4) that are optimized for deep learning tasks, with TensorFlow and Keras APIs for developing the models and hyperparameter tuning [23]. We also validated and tested the model on a local system with an Intel Core i5 (12th Generation) CPU, 16 GB of RAM, 250 GB SSD storage, and an NVIDIA graphics card for model inference, lightweight training, and system-level integration. It integrates computing infrastructure to ensure stability in handling the datasets, faster speed for model training, and repeatability of the performance evaluation across different hardware configurations.

TABLE IV. YOLO Training Configuration

Category	Parameters	Values
Model Settings	Base Model	YOLOv8x
	Task	Object Detection
	Image Size	640 × 640
	Epochs	100
Optimization	Optimizer	SGD
	Initial Learning Rate	0.01
	Momentum	0.937
	Weight Decay	0.0005
Hardware / Device	GPU	NVIDIA(CUDA enabled)
	CPU	Intel Core i5 12th Gen
	RAM	16 GB
	Storage	250 GB SSD
	Workers	8
Data Augmentation	HSV Hue	0.015
	HSV Saturation	0.7
	HSV Value	0.4
	Horizontal Flip	0.5
	Mosaic	1.0
Dataset	Data Configuration	data.yaml

This specific YOLOv8x model was optimized for 100 epochs with 16 images per batch and an input resolution of 640×640 pixels. The optimizer used was SGD with an initial learning rate of 0.01, a momentum of 0.937, and a weight decay of 0.0005. To enhance generalization, data augmentation methods were used, such as HSV color augmentation, horizontal flipping, and mosaic augmentation. All experiments were carried out on an Intel Core i5 (12th Generation) system with 16 GB of RAM and NVIDIA GPU acceleration, as shown in Table IV.

TABLE V. CNN MODEL CONFIGURATION AND TRAINING PARAMETERS

Category	Parameter	Value
Architecture	Backbone	Custom CNN
	Input Size	224 × 224
	Species Classes	5
	Behavior Classes	2
	Dropout	0.5
Optimization	Optimizer	Adam
	Learning Rate	0.001
	Weight Decay	0.0001
Loss	Species Loss	Cross Entropy
	Behavior Loss	Cross Entropy
	λ_1 (Species)	1.0
	λ_2 (Behavior)	1.0
Training	Epochs	100
	Batch Size	16
Hardware	GPU	NVIDIA CUDA
	RAM	16 GB

Table V illustrates that the proposed custom CNN model has an input size of 224 x 224. It has 5 classes for species and 2 classes for behavior. To avoid overfitting, a dropout layer with a rate of 0.5 is used [24,25]. The network is trained with the Adam optimizer, learning rate = 0.001 and weight decay = 0.0001. Both the loss function for the species and the loss function for the behavior classification are cross-entropy with λ_1 = 1.0 for species and λ_2 = 1.0 for behaviour classification. Training is done for 100 epochs, using a batch size of 16. All experiments are run on a system with an NVIDIA CUDA GPU and 16 GB of RAM.

Table VI shows that the CNN model has three convolutional blocks with an input size of 224×224×3. A block consists of a convolutional layer (3×3) and a ReLU activation followed by a max-pooling layer (2×2). There are 32, 64, and 128 filters in the convolutional layers, respectively. The feature maps are then flattened (86,528 units) and fed into a fully connected dense layer with 256 units, with a dropout layer with a rate of 0.5 to avoid overfitting. Finally, two parallel dense layers are used as outputs: species classification (5 classes) and behaviour classification (2 classes), both with Softmax activation. The total number of trainable parameters is 22,246,471 (~84.86 MB).

TABLE VI. PARAMETERS OF MULTI-TASK CNN MODEL

Layer (Type)	Output Shape	Parameters	Description
InputLayer	(224×224×3)	0	Input image
Conv2D (3×3, 32)	(222×222×32)	896	ReLU activation
MaxPooling2D (2×2)	(111×111×32)	0	Pooling layer
Conv2D (3×3, 64)	(109×109×64)	18,496	ReLU activation
MaxPooling2D (2×2)	(54×54×64)	0	Pooling layer
Conv2D (3×3, 128)	(52×52×128)	73,856	ReLU activation
MaxPooling2D (2×2)	(26×26×128)	0	Pooling layer
Flatten	86,528	0	Flatten feature maps
Dense (256)	256	22,151,424	Fully connected layer, ReLU
Dropout	256	0	Dropout 0.5
Dense (5)	5	1,285	Species classification (Softmax)
Dense (2)	2	514	Behavior classification (Softmax)

Total parameters: 22,246,471 (~84.86 MB)
Trainable parameters: 22,246,471
Non-trainable: 0

1) YOLO model benchmarking and selection: YOLOv8 Nano is the most suitable for real-time field surveillance with the fastest inference time, 193.25ms per frame, and the highest FPS, 5.17fps. YOLOv8 Large and YOLOv5 Large have much higher inference times, making them unsuitable for real-time applications. In summary, YOLOv8 Nano provides the closest compromise between accuracy of detection and the speed of the system, and is the ideal choice for the proposed herbivorous animal surveillance framework. As indicated in the below Table VII.

TABLE VII. SELECTED YOLO MODEL METRICS

Model	Pre	Rec	F1-Score	mAP@0.5	mAP@0.5:0.95	Inference Time (ms)	FPS
YOLOv8 Nano	0.8133	0.8159	0.7947	0.5821	0.3198	193.25	5.17
YOLOv8 Large	0.7298	0.7178	0.7144	0.5304	0.2894	2617.14	0.38
YOLOv8 Medium	0.7984	0.7722	0.7723	0.5241	0.2970	861.29	1.16
YOLOv8 Small	0.7917	0.7828	0.7732	0.5724	0.3537	363.09	2.75
YOLOv5 Large	0.7919	0.6683	0.6862	0.4693	0.2679	2574.43	0.39
Yolov5 meduim	0.87142	0.871429	0.861282	0.544692	0.317363	730.0422	1.369784
Yolov5small	0.60094	0.574359	0.501412	0.011558	0.001967	318.9177	3.135605

TABLE VIII. SELECTED MULT-TASK CNN METRICS

Model	Sp.Acc	Sp.Pre	Sp.Rec	Sp.F1	Be.Acc	Be.Pre	Be.Rec	Be.F1	Inference(ms)	FPS
CUSTOMIZECNN	0.9772	0.9775	0.9771	0.9771	0.9814	0.9816	0.9812	0.9814	3.18	314
EFFICIENTNETV1	0.2627	0.1319	0.2498	0.1536	0.4721	0.2361	0.5000	0.3207	6.77	147.7
EFFICIENTNETV2	0.2517	0.2470	0.2457	0.1908	0.5507	0.7050	0.5733	0.4779	2.68	372.7
MOBILENET	0.9949	0.9951	0.9950	0.9950	0.9949	0.9951	0.9947	0.9949	4.06	246.5
RESNET	0.9975	0.9975	0.9975	0.9975	0.9949	0.9950	0.9948	0.9949	3.30	303.3

The best overall performance was the YOLOv8-Nano model, which was chosen as the backbone of the detection network in RAID-Net.

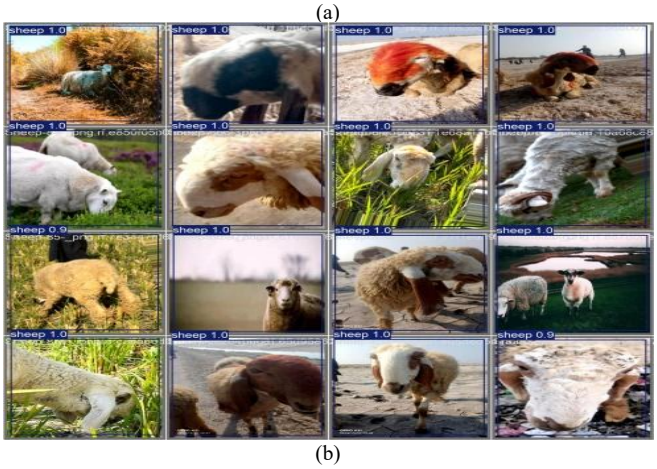
2) Selected multi-task CNN model results: In the multi-task classification experiment, CNN architectures were tested. Table VIII shows that the customized CNN model has the best inference time and frame per second (FPS). The Customized CNN is determined to be the best classification backbone in RAID-Net.

B. Results and Analysis

The image in Fig. 11 (a) shows sample outputs of an animal detection system where different herbivorous animals are identified using bounding boxes. Each box identifies the presence of animals like camel, cow, goat, sheep, and buffalo in various scenes.



(a)



(b)

Fig. 11. (a) The training images; (b) The validation images.

The image in Fig. 11 shows sample outputs of an animal detection system where different herbivorous animals are identified using bounding boxes. Every box identifies the presence of animals like camel, cow, goat, sheep, and buffalo in various scenes.

1) Accuracy comparisons of all yolo models: Fig. 12 is a grouped bar chart that compares the detection accuracy of different livestock species (buffalo, camel, cow, goat, and sheep) using various models of YOLOv5 and YOLOv8, which have different configurations of model sizes (Nano, Small, Medium, and Large). The visualization shows, in comparison, how strong the feature extraction and object localization are in each architecture. It is worth noting that the YOLOv5 Medium model achieves the best performance for these classes in terms of Precision, Recall, and F1-Score, while the YOLOv8 Nano and Small models are extremely competitive in terms of accuracy while having lower model complexity.

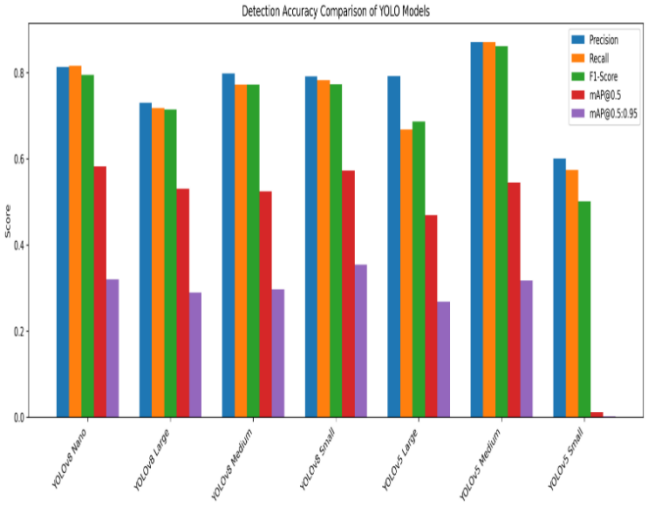


Fig. 12. Detection accuracy comparison.

2) Computational efficiency and speed: Fig. 13 shows the computational efficiency of the tested YOLO models with the help of a dual-axis graph. The primary y-axis (blue bars) represents the inference time for buffalo, camel, cow, goat, and sheep in milliseconds (ms), and the secondary y-axis (the yellow line plot) represents the processing speed (in frames per second: FPS). The figures show clearly that the complexity and the processing speed are inversely related. YOLOv8 Nano not only has the highest FPS and the shortest inference time, but it

also has good deployment potential for live monitoring in the farm.

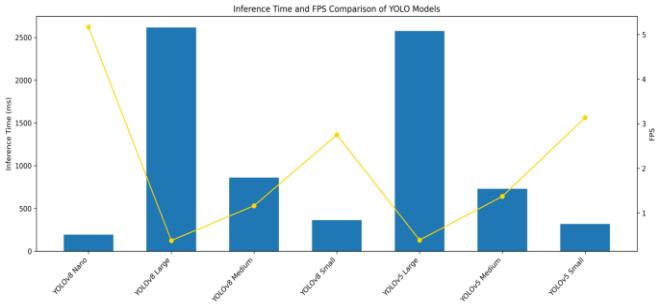


Fig. 13. Computational efficiency and speed.

3) Holistic performance: This radar graph shows a multivariate comparison of the YOLOv5 and YOLOv8 models for buffalo, camel, cow, goat, and sheep detection. The axes range from Precision, Recall, F1-Score, mAP@0.5, mAP@0.5:0.95, Inference_ms, and FPS. The shaded polygons represent the overall performance curve of each model variant, enabling a quick evaluation of the model's detection ability and processing speed. Such a holistic approach is essential to enable the proper selection of the most suitable model architecture that can give high-fidelity identification of objects while maintaining fast inference times required for continuous livestock surveillance. Shown in Fig. 14.

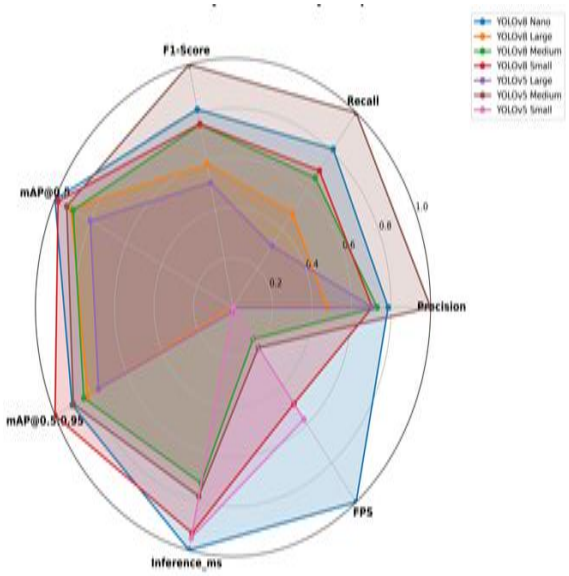


Fig. 14. Holistic performance profiling.

4) CNN model classification performance: These charts demonstrate in Fig. 15 an almost perfect accuracy (close to 1.0) in species and behavior detection of the Custom and ResNet models compared to any of the EfficientNet variants.

5) CNN models inference time and frames per second: This visualization shows that the EfficientNetV2 model has a peak speed of more than 350 FPS, but at the cost of classification accuracy compared to the Custom model, as shown in Fig. 16.

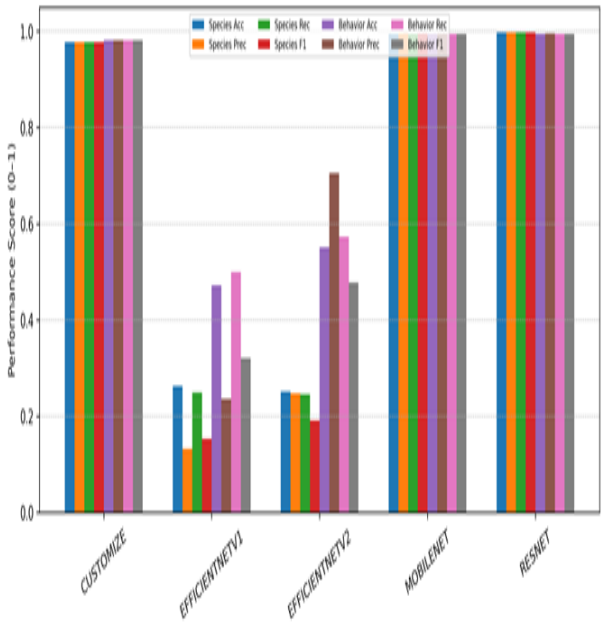


Fig. 15. Classification performance of CNN models.

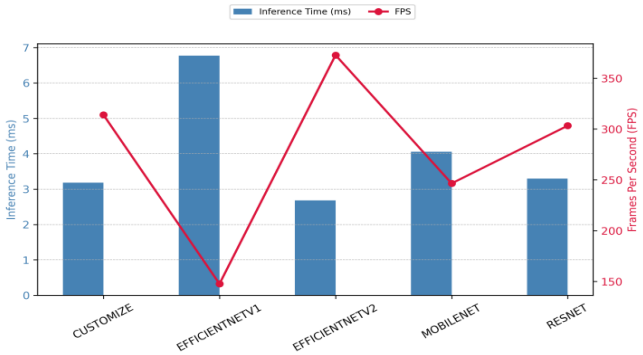


Fig. 16. CNN inference time and FPS.

6) Classification models accuracy and efficiency radar: This summary radar chart demonstrates that overall, the customized models are the most consistent and accurate on all behavior and species metrics. Shown in Fig. 17.

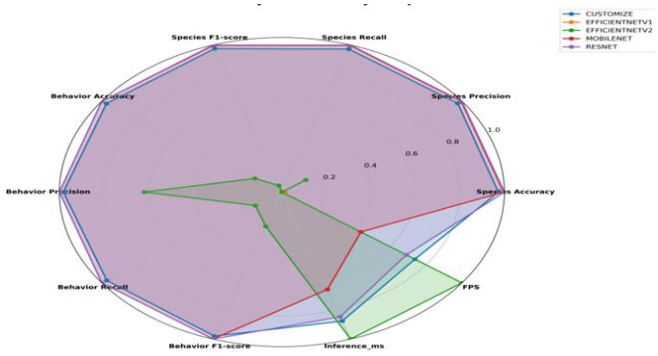


Fig. 17. CNN accuracy and efficiency radar.

7) RAID-Net results: The primary success of the integrated system is compactly summarized in this bar chart: With high Species Accuracy (0.9005) and Behavior Accuracy (0.9431), an

outstanding Alert Precision of 0.9892 is achieved. As shown in Fig. 18, the Customized Multi-task CNN performed 97.72% species classification and 98.14% behaviour recognition accuracy. The complete RAID-Net pipeline, however, has a Hybrid Accuracy of 82.73%, which indicates the overall accuracy of the system at the object detection, ROI localization, classification, and decision-level alert processing stages. Thus, the standalone CNN results are not to be considered the end-to-end accuracy of RAID-Net.

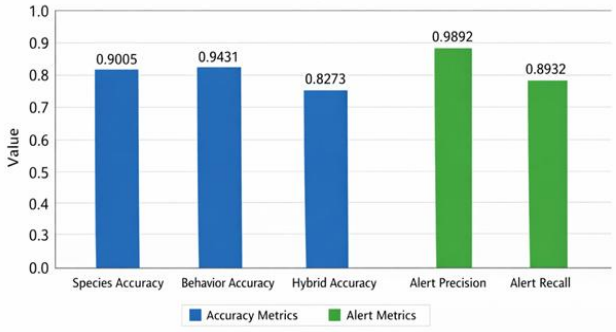


Fig. 18. Performance metrics of the RAID-NET.

Real-time performance is ensured in the system with an average value of 17.61ms and an average FPS of 56.79, suitable for live surveillance applications (Fig. 19).

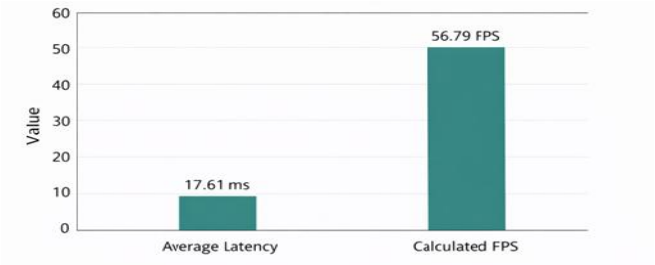


Fig. 19. Latency and FPS performance of the RAID-Net.

This visualization shows the overall performance of your custom framework, showcasing well-balanced high accuracy in hybrid, reliability in alerts, and computational latency.

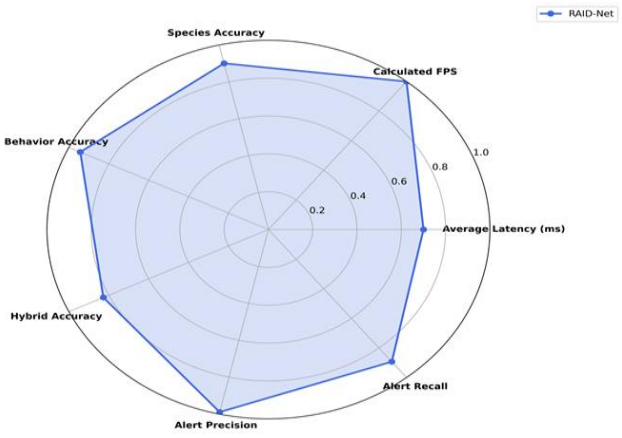


Fig. 20. RAID-Net performance metrics radar chart.

The radar chart shows the overall performance of the RAID-Net model, illustrated in Fig. 20. Species and behavior accuracy are high, and alert precision/recall and computational efficiency are also high.

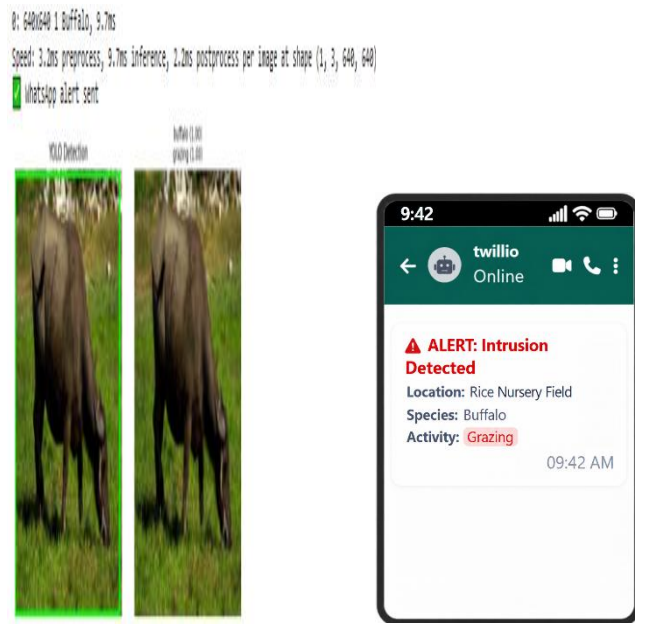
8) RAID-Net performance: The RAID-Net was tested in terms of performance and its feasibility in agricultural applications. The main performance characteristics of the system are summarized in Table IX. The model's Species Accuracy was found to be 90.05%, showing that it was able to identify herbivorous animals with a high level of accuracy, while the Behavior Accuracy was determined to be 94.31%, demonstrating the model's ability to accurately identify animal behavior. The overall Hybrid Accuracy was found to be 82.73%, as a result of the integration of species and behavior recognition

TABLE IX. PERFORMANCE METRICS OF THE RAID-NET

Metric	Value
Average Latency (ms)	17.61
Calculated FPS	56.79
Species Accuracy	0.9005
Behavior Accuracy	0.9431
Hybrid Accuracy	0.8273
Alert Precision	0.9892
Alert Recall	0.8932

These performance metrics are shown in Fig. 18. The accuracy-related metrics Species, Behavior, and Hybrid are represented in blue, while the alert-related metrics Precision and Recall are represented in green. The high Alert Precision of 98.92% indicates that the majority of alerts that the system issues are valid, while the Alert Recall of 89.32% indicates that the majority of relevant intrusion events are detected. The system can be used for real-time monitoring for suitable live agricultural applications with an average latency of 17.61ms and calculated FPS of 56.79.

9) Experimental evaluation of RAID-Net for grazing detection and moving animal tracking: Real-time detection and automated response mechanisms showcase the end-to-end capabilities of the proposed framework. The system successfully identified buffalo that are grazing with a high confidence score and also calculated the inference speed to achieve low-latency performance, as shown in Fig. 21(a). When such an intrusion is detected, the framework automatically sends an SMS to a mobile device through the Twilio API, sending a precise alert to the farm manager over WhatsApp, indicating the location, type of species, and what was seen on the farm. Moreover, the explanation of how the intrusion detection and audio warning system works is given practically in the rice nursery field in Fig. 21(b). In this case, the model detects a goat intrusion and triggers an audible alarm very quickly, which means an immediate physical deterrent to help prevent the crops from being damaged.



(a)



(b)

Fig. 21. (a) Real-time detection and automated WhatsApp alert, (b) Intrusion detection and audio warning system.

V. Ablation Study

To assess the contribution of each component in the RAID-NET, three different configurations were tested: YOLO only, CNN only, and RAID-Net. The summary of key performance metrics for the RAID-NET model is shown in Table VII. Results show the RAID-Net model is always superior to single-component models, giving the highest FPS with high accuracy in species recognition and behavior recognition. Specifically, the hybrid configuration shows that the integration of the detection system based on YOLO with multi-task CNN classification and severity modelling system achieves the Species Accuracy of 90.05%, Behavior Accuracy of 94.31%, and Hybrid Accuracy of 82.73%, respectively, showing efficient inference and reliable recognition.

VI. Limitation

Although the RAID-NET has great potential, some drawbacks still exist. First, the model has been trained and tested largely with daytime data, which may limit its performance at night. Secondly, the current system has no thermal imaging function that would help in detecting people in low-light or obstructed conditions. Lastly, the system has been assessed only in one field test, and the scalability and robustness of the system in various agricultural settings need to be proven. Future studies are needed to address these limitations to improve the versatility and reliability of the model for use in practical farming.

VII. Future Work

These are just some of the future enhancements that are planned for the RAID-NET, and they will help the system become more accurate, scalable, and adaptable. In conclusion, the RAID-NET is a promising solution that has the potential to revolutionize agricultural monitoring in the future. The creation of a vision of the environment and an assessment of the thresholds for confidence. Environmental visual-condition analysis and assessment of thresholds of confidence.

VIII. Conclusion

In this study, the hybrid deep learning-based real-time, behavior-aware herbivorous animal intrusion detection system in rice nursery fields, called RAID-Net, is proposed. The system can detect a variety of species and distinguish their actions by using YOLO-based detection, multi-task CNN classification, and severity modeling, which have high detection accuracy and can effectively reduce false alarms. The outcomes show that RAID-Net is a promising tool to improve the efficiency of agricultural surveillance, giving earlier alerts for crop protection, and reducing labour costs. Moreover, it is an extensible and edge-deployment-friendly framework and offers a smart and feasible protection solution for modern agriculture. Overall, RAID-Net is a strong artificial intelligence-based solution to protect rice nursery fields from herbivorous intrusion, which will help achieve sustainable and technology-driven agriculture.

References

[1] Çimen, M. E. (2025). Comparison of Deep Learning and YOLOv8 Models for Fox Detection Around the Henhouse. Journal of Smart Systems Research, 5(2), 76-90.

[2] Liu, Li, G., Sun, J., Guan, M., Sun, S., Shi, G., & Zhu, C. (2022). A New Method for non-destructive identification and Tracking of multi-object behaviors in beef cattle based on deep learning. Animals, 14(17), 2464.

[3] Chappidi, J., & Sundaram, D. M. (2024). Novel Animal Detection System: Cascaded YOLOv8 with Adaptive Preprocessing and Feature Extraction. IEEE Access.

[4] Othman, S., Mavani, N. R., Hussain, M. A., Abd Rahman, N., & Ali, J. M. (2023). Artificial intelligence-based techniques for adulteration and defect detection in the food and agricultural industry: A review. Journal of Agriculture and Food Research, 100590.

[5] Albahar, M. (2023). A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities. Agriculture, 13(3), 540.

[6] Mesías-Ruiz, G. A., Pérez-Ortiz, M., Dorado, J., de Castro, A. I., & Peña, J. M. (2023). Boosting precision crop protection towards agriculture 5.0

via machine learning and emerging technologies: A contextual review. Frontiers in Plant Science, 14.

[7] Dracatos, P. M., Lück, S., & Douchkov, D. K. (2023). Diversifying resistance mechanisms in cereal crops using microphenomics. Plant Phenomics, 5, 0023.

[8] Ezugwu, A. E., Oyelade, O. N., Ikotun, A. M., Agushaka, J. O., & Ho, Y. S. (2023). Machine Learning Research Trends in Africa: A 30-Year Overview with Bibliometric Analysis Review. Archives of Computational Methods in Engineering, 1-31.

[9] Vidhyalatha, T., Sreeram, Y., & Purushotham, E. (2022). Animal Intrusion Detection using Deep Learning and Transfer Learning Approaches. International Journal of Human Computations & Intelligence, 1(4), 51-60.

[10] Ukwuoma, C. C., Qin, Z., Yussif, S. B., Happy, M. N., Nneji, G. U., Urama, G. C., ... & Agobah, H. (2022). Animal species detection and classification framework based on modified multi-scale attention mechanism and feature pyramid network. Scientific African, 16, e01151.

[11] Gowda, T., Sahana, R., Raksha, H., Bindushree, R. A Review on Animal Encroachment Detection in Crop Fields Using Deep Learning and IoT.

[12] Sandeep, T., Manushree, B., Rahul, S., & Bharath, T. (2022). Crop Protection from animals using CNN.

[13] Fu.Wang, Zhang, C., Xu, P., & Ruan, W. (2022). Deep learning and its adversarial robustness: A brief introduction. In Handbook On Computer Learning And Intelligence: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation (pp. 547-584).

[14] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

[15] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[16] Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "Efficientdet: Scalable and efficient object detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[17] Lin, Tsung-Yi, et al. "Focal loss for dense object detection (pp. 1)." IEEE transactions on pattern analysis and machine intelligence (2018).

[18] Dai, Jifeng, et al. "Deformable convolutional networks." Proceedings of the IEEE international conference on computer vision. 2017.

[19] Gu, H., Wang, Y., Hong, S., & Gui, G. (2019). Blind channel identification aided generalized automatic modulation recognition based on deep learning. IEEE Access, 7, 110722-110729.

[20] Jhatial, M. J., Shaikh, R. A., Shaikh, N. A., Rajper, S., Arain, R. H., Chandio, G. H., & Shaikh, K. H. (2022). Deep Learning-Based Rice Leaf Diseases Detection Using Yolov5. Sukkur IBA Journal of Computing and Mathematical Sciences, 6(1), 49-61.

[21] Zhang, X., Zhou, X., M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6848-6856)

[22] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[23] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[24] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.

[25] Sana, S., Gudla, S., Kandi, H., Mishra, P., Sambana, B., Hameed, S. S., & Ramesh, Y. Amalgam enhanced to Computer Vision-Deep Learning and Internet of Things for Aerial View Cultivating Monitoring. MMTC Communications-Frontiers.

Interpretable Deep Convolutional Neural Network Framework for Online Reputation Management Using Customer Reviews and Social Media Feedback

An Interpretable Deep Learning Approach for Online Reputation Management

Abdullah Saad AL-Malaise[1]*, Bandar A. Alrami Alghamdi[2], Mahmoud Ragab[3]
Faculty of Computer Studies, Arab Open University, Riyadh, Saudi Arabia[1, 2]
Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia[3]

***Abstract*—Social media has become a significant channel for communication and interaction, significantly influencing how organizations are perceived in the digital era. Online reputation management (ORM), as an addition to public relations, concentrates on shaping and maintaining an organization's image through continuous surveillance of customer reviews and online feedback. Traditional methodologies, such as machine learning and deep learning–based reputation management, depend upon learned patterns and predetermined features from historical information; recent ORM approaches exploit real-world data from social platforms to extract dynamic public sentiment. Meanwhile, online opinions can transform rapidly because of human-generated text, and preserving a reliable reputation has become progressively difficult for organizations. To improve interpretability and transparency, explainable AI algorithms are frequently incorporated into recent ORM architectures. This study presents an Interpretable Deep Convolutional Neural Network Framework for Online Reputation Management (IDCNN-ORM). The proposed model integrates an improved mRMR approach to identify highly informative and non-redundant features. Moreover, the deep convolutional neural network is applied to learn hierarchical feature representations from input data. The Adabelief optimizer is integrated into the proposed framework to enhance training stability by adaptively refining the learning process. Furthermore, the LIME-based explainability artificial intelligence technique is utilized to enhance transparency in the model's decision-making process. The experimental analysis of the proposed IDCNN-ORM model is carried out using the Social Media Sentiments Analysis Dataset. The comparative analysis demonstrates promising performance with an accuracy of 94.85% compared with existing methodologies.**

***Keywords*—Online reputation management; social media feedback; adabelief optimizer; explainable artificial intelligence; deep convolutional neural network**

I. Introduction

Social media networks have developed into an important platform where customers can express their sentiments, engage with brands, and share their experiences in the modern world [1]. The velocity and volume of user-generated content (UGC) on applications like LinkedIn, Facebook, YouTube, Twitter, and Instagram offer a valuable data resource that can be used for several marketing purposes.

*Corresponding author.

Among these, Online reputation management (ORM) is the process of improving, tracking, observing, and handling an individual's or organization's reputation in the digital era [2]. Nowadays, UGC plays an essential part in influencing public perception through blog comments, feedback, ratings, posts, forums, and reviews. Although UGC has enhanced data sources, it is difficult to extract useful and meaningful insights from the volume and the considerable difficulty [3]. Reading and analyzing the sentiment of all relevant online texts requires huge effort and time, and it is frequently quite subjective. Experts struggle with a huge amount of information, unreliable language, and complex emotional interpretation [4]. This type of interpretation is ineffective and restricts a firm's capability to gather real-world insights, respond quickly to novel issues, and evaluate opinions through large-scale databases or prolonged periods. Inefficient processing of data streams can result in a failure to understand consumer requirements, enhance promotional measures, and prevent damage to brand reputation [5].

Over the past few decades, sentiment analysis (SA) has evolved as a vital domain within machine learning (ML) and natural language processing (NLP), driven by the explosive growth of digital information [6]. Given the applications of social media networks, customer opinion, online reviews, and feedback, comprehending the sentiments extracted in a text is important [7]. SA, termed opinion mining, is the computing operation of classifying, mining, and evaluating subjective data from a text or content to determine the complete attitude or sentiment expressed toward specific products, entities, events, or topics [8]. This method includes identifying text into pre-defined classes like neutral, positive, and negative sentiments, or more refined opinions and emotions. Usually, ML-driven SA methods are extended, and novel gaps are classified by SA identification. There is a move toward using more intricate methods to solve this limitation [9]. Deep learning (DL) models are a branch of the ML framework that provide effective solutions to these tasks with structures that involve recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM), indicating efficiency in managing intricate SA challenges. These methods have gained attention for their capability of capturing the contextual interaction within text and have exhibited better performance compared with other DL models in sentiment analysis tasks [10].

Online Reputation Management (ORM) has become important owing to the huge increase in social media feedback and customer comments on digital platforms. Organizations need an automatic strategy to examine a huge amount of noisy and unstructured textual data in the real world. The Kaggle social media sentiment analysis dataset signifies real-time sentiment demonstrations from distinct online sources. Conventional ML systems frequently fail to extract semantic relationships and deep contextual information in certain cases. Therefore, there is a need for innovative DL-based techniques to enhance feature representation and sentiment classifier outcomes in ORM models. This motivates the proposed Interpretable Deep Convolutional Neural Network Framework for Online Reputation Management (IDCNN-ORM) using social media data.

- Utilize an improved mRMR approach to effectively select the most relevant and non-redundant features from the dataset.
- Design a deep convolutional neural network (DCNN) approach to learn hierarchical feature representations from input data by continuously capturing complex patterns at multiple levels to improve classification performance.
- Present an Adabelief optimizer for adaptively modifying learning rates, thereby ensuring stable and efficient training.
- Integrate a LIME-based explainable artificial intelligence (XAI) model for providing interpretability and improving transparency in the decision-making process.
- Evaluate the performance of the proposed model using the Social Media Sentiments Analysis Dataset, demonstrating its effectiveness under various evaluation measures.

II. Literature Review

This section concisely reviews the existing research on online reputation management. Semenda et al. [11] discussed the objective of matrix-based examination and game theory in improving social media marketing algorithms for e-commerce industries. To integrate these mathematical techniques with social media analytics, the goal is to deliver a comprehensive architecture that can predict customer action. This system allows sellers to organize more adaptive, targeted, and efficient marketing campaigns, drive growth, and improve customer fulfillment in the digital marketplace. Istanbulluoglu and Sakman [12] examined the associations among business responses to customer repurchase intentions and social media complaints. The connection between purchase intention and the five scopes of handling complaints, as led by customer trust in the corporation, was surveyed, and users' propensity to trust was hypothesized as a mediator in this correlation. Saheb et al. [13] examined the connection between social media and AI. The study employs a mixed-model strategy to examine 1540 scholar records, incorporating systematic literature review and bibliometric approaches. The major objective of this study is to detect the most significant themes and trends, and also the impact of company values and suggestions, in the AI Social Media field. The initial level of the study encompassed a quantitative keyword analysis, which led to the identification of ten key themes.

Alsemaree et al. [14] offer the requirement for a nuanced consideration of consumer sentiments under several coffee goods, allowing corporations to generate informed decisions concerning improvement, discontinuation, and product promotion. Antczak [15] discusses the major goals for exploring the connections among social media marketing, digital marketing, and customers' online purchasing actions. The study will investigate how marketing actions in social media and digital media, namely social media advertising, product reviews, influencer marketing, and e-mail marketing, impact user behavior and purchase choices. The authors [16] proposed a systematic literature review (SLR) investigating the role of SMIs in the e-commerce market. This study examines recent knowledge on how influencer authority, demarcated by expertise, trustworthiness, and reliability, impacts user behavior, purchase intention, and brand engagement. Odionu et al. [17] introduced the use of big data analytics for enhancing CRM engagement and retention approaches. The main goal is to examine how big data analytics can be exploited to gain insights into consumer actions, interactions, and preferences, thus enhancing CRM proposals. This study contains a comprehensive literature review and case study examination of organizations that have effectively incorporated big data analytics into their CRM methodology.

Existing sentiment analysis systems for ORM are frequently limited in interpretability, making their forecasting complex to trust in real applications. Several strategies fail to efficiently manage noisy and informal social media content, namely, emojis, abbreviations, and slang. The usage of CNN-based frameworks to capture hierarchical textual attributes is still unexplored in comparison with DL algorithms. Furthermore, more research focuses on accuracy without combining explainability methodologies. There is a limited number of unique architecture that integrates higher results with method transparency.

III. Methodological Framework

The proposed IDCNN-ORM model follows a comprehensive pipeline for improving a reputation system using social media data. Fig. 1 shows the overall architecture of the proposed model. As depicted in Fig. 1, the methodology involves five major stages: data preprocessing, feature selection, classification, parameter tuning, and explainability analysis. These processes are integrated to ensure effective model performance and accurate sentiment-based reputation analysis.

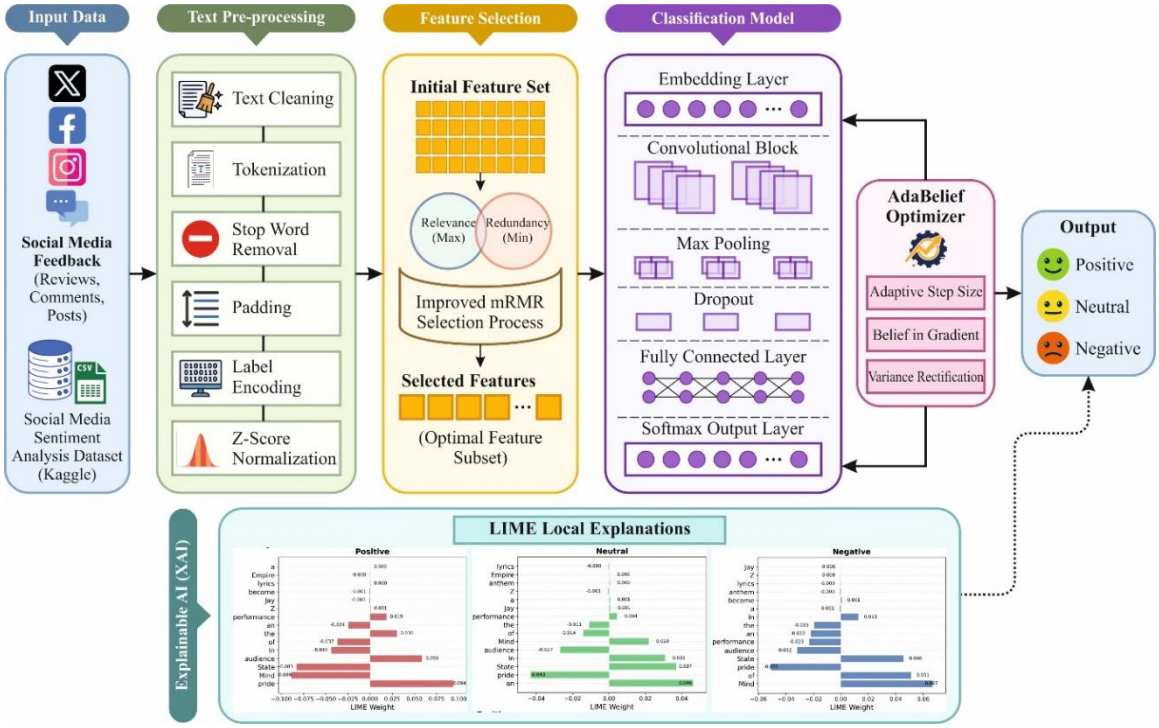


Fig. 1. Overall architecture of the IDCNN-ORM model.

A. Data Preprocessing

A multi-stage preprocessing stage includes text cleaning, tokenization, stop-word removal, and padding to convert raw data into a structured format [18].

1) Text cleaning: It eliminates hashtags symbols (#), URLs, punctuation, emojis, and user mentions from the "Text" column, and special characters. It ensures the preservation of significant expressive words from social media posts.

2) Tokenization: It divides content into smaller units like subwords or words referred to as tokens, while handling emojis, hashtags, special characters, and URLs to extract meaningful phrases for examination.

3) Stop-word removal: It removes common words like “the”, “and”, and “is” that do not enhance meaning. This helps to lessen data dimensionality and increases processing efficiency.

4) Padding: It is used to pad token-based post sequences to guarantee a unified input length for every instance. This is necessary for the DL strategy; later posts in the dataset are different in structure and length.

Before model training, the proposed IDCNN-ORM framework performs text cleaning, tokenization, stop-word removal, padding, label encoding, and Z-score normalization to preprocess the social media text data.

5) Label encoding: This encoding approach is leveraged to convert categorical variables into numerical form. It allocates a unique integer value to every category, allowing effective learning from categorical data.

Z-score normalization (standardization) converts information; thus, it has a standard deviation of 1 and a mean of 0. During training, it is utilized to scale each attribute to a standard scale, which improves system performance, faster convergence, and stability.

$$X_{norm} = \frac{X - \mu}{\sigma} \tag{1}$$

Here, μ means the mean of the feature, X signifies the actual feature value, X_{norm} indicates the scaled value, and σ implies the standard deviation of the feature.

B. Improved mRMR-Based Feature Selection

An improved mRMR-based feature selection (FS) strategy is applied to select the most relevant and non-redundant attributes from the dataset. It reduces feature redundancy and computational complexity while improving the effectiveness of the classification process [19]. The standard mRMR is an effective FS strategy that repeatedly increases the number of attributes in the feature subset by a gradual search that balances the dual terms: minimal redundancy to the gathered feature subset and maximum relevance to the label. The dual relevance is computed.

$$\max A(S, C) = \frac{1}{n} \sum_{f_i \in S} I(f_i, C) \tag{2}$$

$$\min R(S) = \frac{1}{n^2} \sum_{f_i f_j \in s} I(f_i, f_j) \tag{3}$$

here, C signifies the data label, S denotes the chosen feature subset. $A(S, C)$ refers to the correlation between the chosen subset and the label, f_i and f_j indicate the dual attributes

features in S, $I(*)$ represents the mutual information, and $R(S)$ implies the redundancy among the chosen subset.

$$I(X,Y) = \int_X \int_Y P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)} \quad (4)$$

As an instance, $P(X,Y)$ refers to their marginal probability distributions. Conversely, its mutual data-driven importance assessment is more appropriate for diverse information. These dual behaviors can be examined to manage continuous data, and it has been established that they can deliver information. Pearson correlation and T-test coefficient equations are defined.

$$t(f_i) = \frac{\left|\overline{f_{i_{pos}}} - \overline{f_{i_{neg}}}\right|}{\sqrt{\frac{S^2_{i_{pos}}}{n_{pos}} + \frac{S^2_{i_{neg}}}{n_{neg}}}} \quad (5)$$

$$\rho(f_i, f_j) = \frac{\sum (f_i - \overline{f_i})(f_j - \overline{f_j})}{\sqrt{\sum (f_i - \overline{f_i})^2 \sum (f_j - \overline{f})^2}} \quad (6)$$

Here, $\overline{f_{i_{neg}}}$ implies the mean value of f_i in the negative instance; $\overline{f_{i_{pos}}}$ represents the mean value of the positive instance, n_{pos} and n_{neg} indicate the sample counts; $S^2_{i_{pos}}$ and $S^2_{i_{neg}}$ refer to the variance of f_i.

$$f_i \max \left[t(f_i) - \frac{1}{S} \Sigma f_j \in F_s \rho(f_i, f_j) \right] \quad (7)$$

Additionally, the stopping principle of the standard mRMR methodology for increasing the count of chosen feature subgroups, which needs to be set analytically, generates limitations to ensure the outcome of the last chosen feature subgroups.

C. Online Reputation Classification via DCNN

A deep convolutional neural network is used to analyze sentiment patterns in social media data for effective reputation assessment. The model utilizes deep convolutional layers to learn hierarchical feature representations and improve classification [20]. A neural network (NN) is an effective method comprising several interrelated nodes; here, every node uses an activation function to produce an output. The relationships among nodes carry weights that verify how intensely data is distributed among neurons and serve as the network memory. The output of the network, based on these weights, is learned, and correlations are structured. During Online Reputation Management (ORM), the major features impacting reputation include opinion strength, sentiment polarity, variations in customer reviews, and user engagement and social media comments. Conversely, these components cannot exactly signify the last reputation category, namely neutral, positive, or negative, creating a classification problem utilizing raw information.

The actual reputation grade is derived from consumer feedback and social media text, which precisely imitates public opinion. Hence, these data sources are employed as key inputs for classifying reputation into multiple classes. The DCNN progress comprises capturing attributes from the data and upgrading system hyperparameters through learned extractions. These learned hyperparameters are assigned to different DCNN sub-methods to reduce computation, while preserving results. The n DCNN sub-models for learning hyperparameters are represented by.

$$\theta^S = \left\{ \theta_1^S, \theta_{2,}^S, \ldots, \theta_n^S \right\} \quad (8)$$

where, each θ_n^S corresponds to one sub-model, and θ^S denotes the combined parameters. All sub-models encompass three fully connected layers and five convolution layers to study patterns from social media text and consumer feedback. This transfer learning system assists in transferring knowledge from a source dataset to the targeted ORM classifier study, minimizing the requirement for huge labeled data while sustaining superior precision. Subsequently, transfer learning, n DCNN-TL sub-models are designed, and an ensemble system is created utilizing these sub-models together with one classification layer and one fully connected layer.

The convolution layer captures significant features by employing filters across the input text extraction. The convolution output at position (i,j) for the k_{th} filter.

$$\begin{cases} Z^{l_{conv}}_{i,j,k} = C(X,K)_{i,j,k} = \sum_{r=1}^{k_h} \sum_{s=1}^{k_w} \sum_{t=1}^{k_c} x_{i',j',t'} \, k_{r,s,t',k} + b_k \\ i' = (i-1)S_h + r \\ j' = (j-1)S_w + s \end{cases} \quad (9)$$

In Eq. (9), bk signifies the bias term, and k denotes the filter weights. The convolution layer helps to capture suitable patterns from text, including contextual meaning and sentiment words. These features are applied to the classifier for reputation as neutral, negative, or positive.

The method repeatedly updates the feature extraction depending on observed information. During training, it improves forecasting by comparing predicted and actual class labels and reducing errors.

A major benefit of DCNN is that it can manage variable-length text information from comments and social media without strict input size challenges. It acquires valuable patterns from noisy and unorganized text data and transforms them into essential features for classification.

$$q(\chi_k | x_{0:k}(i), y_{1:k}) = p\left(x_k | \chi_{k-1}(i)\right) \quad (10)$$

During learning, this supports the model in concentrating on the most essential instances. If the count of efficient records becomes too small, then the algorithm is restarted to maintain results. Afterward, resampled, a novel group of attributes is formed and employed for the final classification.

$$w_t^{(i)} = (w_{t-1^{(i)}})^{\alpha} \frac{p(z_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1^{(i)}})}{q(x_t^{(i)} | x_{t-1}^{(i)}, z_{1.t})} \quad (11)$$

where, α influences how strongly weights are upgraded. Finally, the system is evaluated on the basis of detection precision and computational effectiveness. An added balancing parameter is applied to control imbalance among sentiment

categories. The proposed techniques efficiently classify online reputation from user feedback and social media information, utilizing ensemble transfer learning and DCNN-based feature learning. Fig. 2 denotes the overview of the DCNN model for reputation classification.

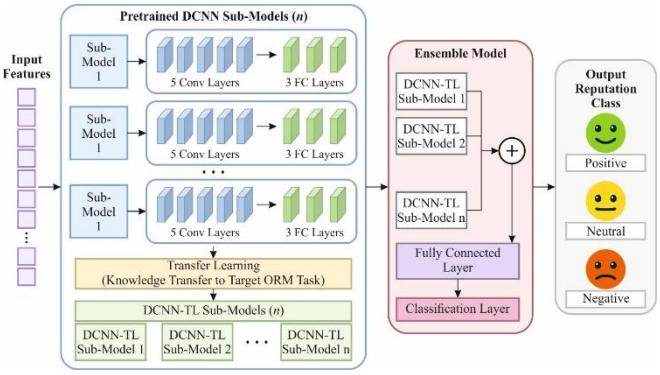


Fig. 2. Overview of the DCNN model for reputation classification.

D. Hyperparameter Optimization

AdaBelief optimizer is an adaptive optimization algorithm that adjusts learning rates based on the difference between predicted and observed gradients [21]. Adaptive optimizers, namely Adam, converge more quickly than SGD; however, their consistency and outcome might not be sufficient for larger datasets. This optimizer proposes an alteration to the Adam optimization for improving model generalization and training stability. During Adabelief's dissimilar Adam, a novel variable s_t is defined as the exponential moving average of $(g_t - m_t)^2$, while $g_t - m_t$ denotes the deviation. A smaller variance amongst g_t and m_t refers to a robust belief, while in the case of a weaker belief, the transformation is smaller. According to this methodology, the Adabelief measures s_t and biased correction $\widehat{s_t}$ than v_t and $\widehat{v_t}$,

$$s_t = \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2 + \varepsilon, \tag{12}$$

with its bias-corrected form represented by:

$$\widehat{s_t} = \frac{s_t}{1-\beta_2^t} \tag{13}$$

and the AdaBelief upgrade principle has been illustrated by

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{s}}+\in}\widehat{m_t} \tag{14}$$

AdaBelief leveraged the same parameter settings as Adam, with identical default values.

E. Model Interpretability

LIME is an extensively used technique in the field of XAI, which intends to deliver transparent explanations for independent forecasting of intricate ML approaches [22]. It works on the principle that it can be approximated more easily by explainable strategies. Therefore, a model-agnostic system, LIME, is developed across a wide range of classifications and regressions without demanding access to the internal pattern of the algorithm. The major goal is to produce local approximators of a system's action in the specified occurrence to construct an interpretable method, usually a linear regression strategy.

$$\mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{15}$$

Here, g represents a group of simpler approaches, $\pi_x(z)$ refers to a proximity among z and the χ, $\mathcal{L}(f, g, \pi_x)$ denotes the fidelity of g.

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right) \tag{16}$$

where, σ regulates the width of the neighborhood, and $D(x, z)$ implies a distance metric, $\Omega(g)$ represents a complexity penalty.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z) \cdot (f(z) - g(z))^2 \tag{17}$$

Here, Z signifies a group of perturbed instances in χ.

IV. Performance Analysis and Discussion

The experimental assessment of the IDCNN-ORM technique is achieved using the Social Media Sentiments Analysis Dataset [23]. This dataset captures a vibrant tapestry of emotions, tendencies, and relationships across diverse social media platforms. This dataset offers images of user-generated content, hashtags, combined text, timestamps, retweets, likes, and nations. Every entry reveals unique stories, namely moments of admiration, excitement, surprise, thrill, contentment, and many more exchanged by individuals globally. The dataset comprises three classes: positive, negative, and neutral, as denoted in Table I and Table II, which demonstrate the sample social media content used for sentiment analysis.

TABLE I. Details of Dataset

Class Label	Count
Positive	455
Neutral	112
Negative	165
Total	732

TABLE II. Sample Content

S.no	Text	Sentiment
1	Reflecting on the past and looking ahead.	Positive
2	Trying out a new dinner recipe tonight.	Neutral
3	Traffic was terrible this morning.	Negative

TABLE III. Sentiment Analysis of the IDCNN-ORM Method with 70% of TRAIP

Classes	$Accur_y$	$Preci_n$	$Recal_l$	F1	ROC_{AUC}	MCC
TRAIP (70%)						
Positive	91.21	91.34	95.03	93.15	89.88	81.03
Neutral	96.68	91.04	84.72	87.77	91.68	85.93
Negative	91.02	82.73	77.12	79.82	86.15	74.13
Average	92.97	88.37	85.62	86.92	89.24	80.36

Table III and Fig. 3 present the sentiment analysis of the IDCNN-ORM approach with three classes for 70% of TRAIP. The proposed IDCNN-ORM method achieved an average value with $accur_y$ of 92.97% and MCC of 80.36%, implying its effectiveness as a social media sentiment classifier model.

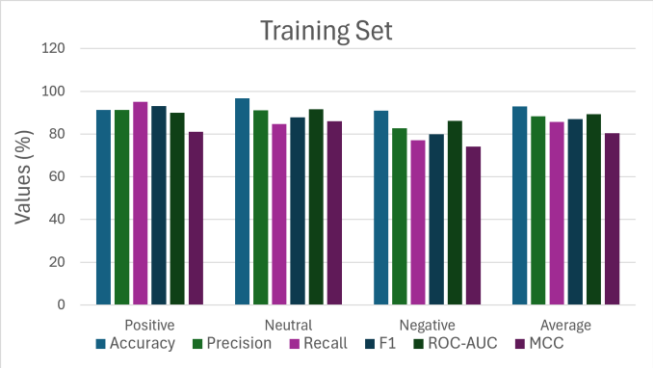


Fig. 3. Performance of the IDCNN-ORM approach with 70% of the training set.

Table IV and Fig. 4 display the sentiment analysis of the IDCNN-ORM method with three classes for 30% of TESTP. The proposed IDCNN-ORM approach achieved an average value of $accur_y$ of 94.85% and an MCC of 86.09%, denoting its efficacy in the social media sentiment classification task.

TABLE IV. SENTIMENT ANALYSIS OF THE IDCNN-ORM METHOD WITH 30% OF TESTP

Classes	**$Accur_y$**	**$Preci_n$**	**$Recal_l$**	F1	**ROC_{AUC}**	MCC
TESTP (30%)						
Positive	93.18	93.38	95.49	94.42	92.57	85.69
Neutral	96.36	92.11	87.50	89.74	92.92	87.58
Negative	95.00	89.13	87.23	88.17	92.17	85.01
Average	94.85	91.54	90.07	90.78	92.55	86.09

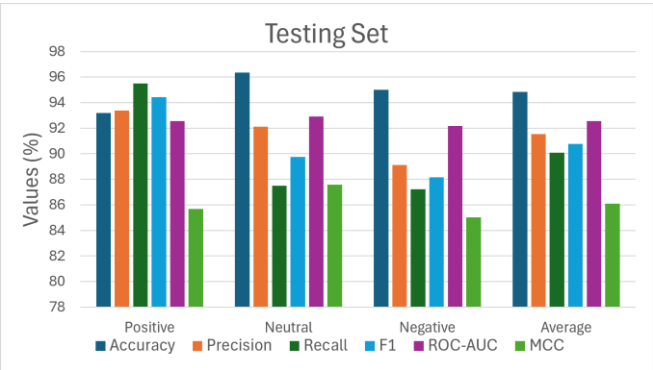


Fig. 4. Performance of the IDCNN-ORM approach with 30% of the testing set.

Fig. 5 depicts the training (TRAING) and validation (VALIDN) $accur_y$ curves of the IDCNN-ORM technique. The TRAING $accur_y$ continually increases, indicating effective learning and correct convergence of the system. The VALIDN $accur_y$ follows the training curve with small differences, suggesting higher generalization performance and lower overfitting.

Fig. 6 exemplifies the TRAING and VALIDN loss curves of the IDCNN-ORM method. The losses decline quickly in the earlier epochs and then decrease smoothly, representing effective learning and appropriate convergence. The minor differences amongst the TRAING and VALIDN curves exhibit less overfitting and superior generalization outcomes.

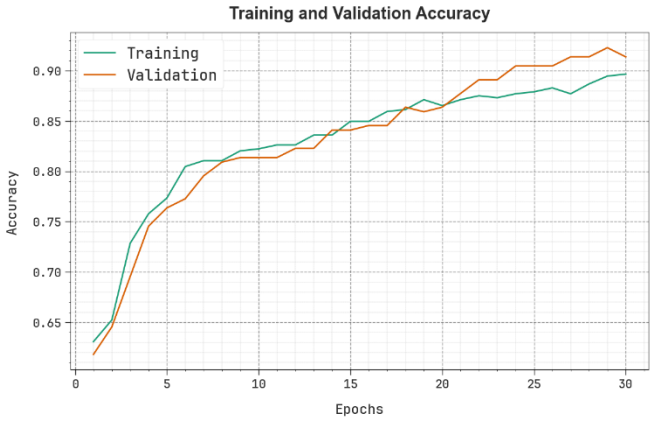


Fig. 5. $Accur_y$ curve of the IDCNN-ORM method.

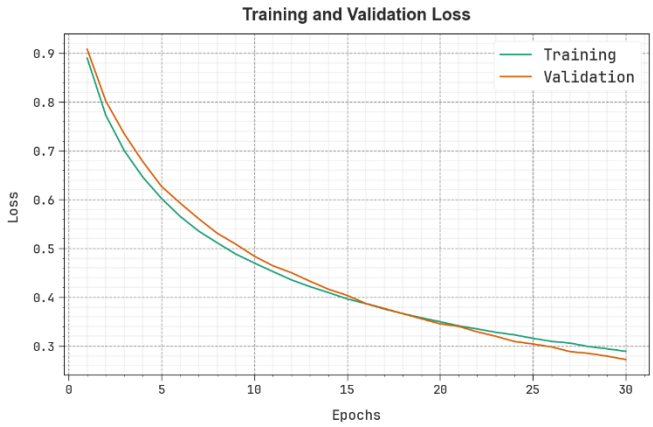


Fig. 6. Loss curve of the IDCNN-ORM approach.

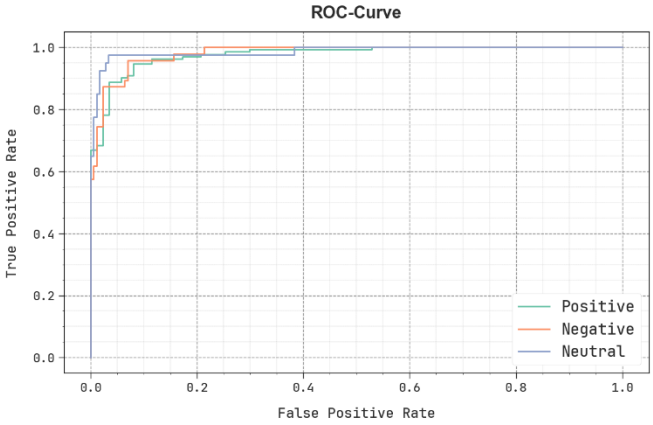


Fig. 7. ROC curve of the IDCNN-ORM method.

Fig. 7 portrays the ROC curve evaluation of the IDCNN-ORM system, representing boosted performance across all classes, signifying its strong discriminative ability. The consistently higher ROC values indicate the efficiency of the model in differentiating among class boundaries, and this steady pattern across numerous categories assures its consistency and robustness in the classification process.

Table V and Fig. 8 depict the comparison outcome of the IDCNN-ORM approach with existing methodologies using evaluation metrics [24-26]. The results imply that the SVM, LSTM, and DistilBERT approaches have lower performance, whereas the RoBERTa, CNN, and RNN methods have slightly greater values. Moreover, the CNN–LSTMHybrid has obtained moderate performance. At the same time, the proposed IDCNN-ORM approach has achieved superior performance with $accur_y$ of 94.85%. $preci_n$ of 91.54%, $recal_l$ of 90.07%, and F1 of 90.78%, emphasizing its robustness of the classifier tasks.

TABLE V. COMPARATIVE RESULTS OF THE IDCNN-ORM METHOD WITH OTHER APPROACHES

Methods	$Accur_y$	$Preci_n$	$Recal_l$	F1
SVM	70.04	71.52	70.4	70.14
CNN	87.62	85.5	88.5	88.7
RNN	87.02	88.1	88.3	87
LSTM	70.66	70.34	69.92	68.45
RoBERTa	80.61	80.54	80.01	80.92
DistilBERT	79.34	79.93	79.14	79.63
CNN–LSTMHybrid	92.14	89.64	87.53	88.32
IDCNN-ORM	94.85	91.54	90.07	90.78

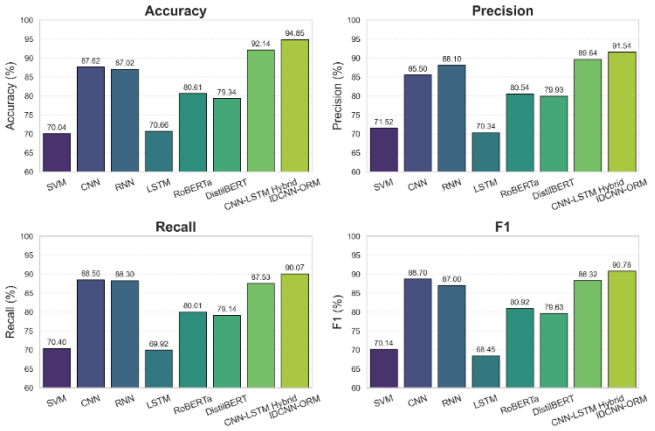


Fig. 8. Comparative results of the IDCNN-ORM method with various metrics.

Fig. 9, LIME-based local explanation of the proposed model's predictions for negative, neutral, and positive reputation classes. This figure indicates the contribution of individual features, where positive LIME weights support the predicted class and negative weights oppose it. Higher absolute weights indicate greater influence on the model's decision.

The proposed IDCNN-ORM architecture exhibits robust solutions in sentiment analysis for the ORM study. The improved mRMR feature selection successfully minimizes repetition and enhances the quality of the input features. The deep convolutional framework extracts rich hierarchical and contextual data from social media content. The AdaBelief optimizer accelerates convergence and improves training stability, enhancing learning efficacy. The experimental outcome demonstrates that the method attains promising outcomes over conventional systems, validating its efficiency.

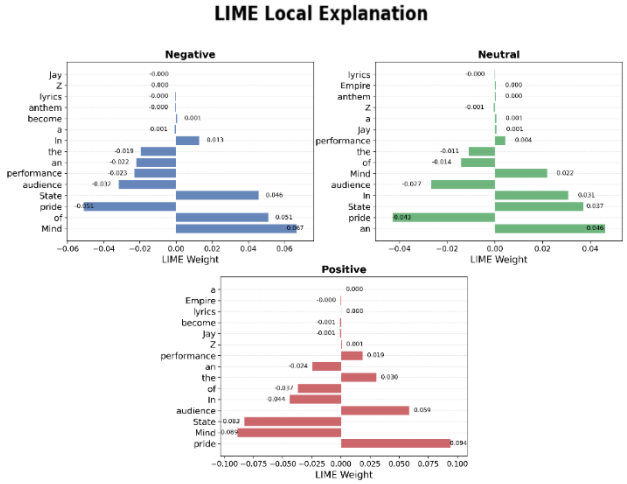


Fig. 9. LIME local explanation of feature contributions for negative, neutral, and positive reputation classifications.

V. CONCLUSION AND FUTURE WORK

This research introduced an IDCNN-ORM approach for enhancing reputation analysis using social media data. The proposed model incorporates an improved mRMR approach to select highly informative and non-redundant features. Moreover, the DCNN is employed to learn hierarchical feature representations from input data. The Adabelief optimizer is applied to enhance training stability by adaptively refining the learning process. Additionally, the LIME-based XAI technique is utilized to enhance transparency in the model's decision-making process. The experimental result analysis of the proposed IDCNN-ORM model is performed using the Social Media Sentiments Analysis Dataset. The comparative analysis demonstrates promising performance with an accuracy of 94.85% compared with existing methodologies. Despite the strong performance of the IDCNN-ORM approach, it still has challenges in handling extremely ambiguous and contextual sentiments in social media content. In the future, it can concentrate on combining transformer-based frameworks to extract long-range contextual dependencies in text information. An innovative feature selection algorithm can be investigated to minimize repetition and enhance adaptability. This technique can be applied to sentiment analysis for wider applicability across global social media platforms. Moreover, integrating progressive explainable AI methodology can improve transparency and customer trust in real-time ORM organizations.

Data Availability Statement: The data that support the findings of this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/kashishparmar02/social-

media-sentiments-analysis-dataset?select=sentimentdataset.csv, reference number [23].

ACKNOWLEDGMENT

The authors extend their appreciation to the Arab Open University for funding this work through research fund No. (AOUKSA-524008).

REFERENCES

[1] Akpuokwe, C.U., Chikwe, C.F. and Eneh, N.E., 2024. Innovating business practices: The impact of social media on fostering gender equality and empowering women entrepreneurs. Magna Scientia Advanced Research and Reviews, 10(2), pp.032-043.

[2] Vale, L. and Fernandes, T., 2025. Social media and sports: driving fan engagement with football clubs on Facebook. In Contemporary Issues in Sports Marketing (pp. 37-55). Routledge.

[3] da Silva Pinho, A., Céspedes Izquierdo, V., Lindström, B. and van den Bos, W., 2024. Youths' sensitivity to social media feedback: A computational account. Science advances, 10(43), p. eadp8775.

[4] Banerji, R. and Singh, A., 2024. Do social media marketing activities promote customer loyalty? A study on the e-commerce industry. LBS Journal of Management & Research, 22(1), pp.93-109.

[5] Muftah, M., 2024. Impact of social media on learning the English language during the COVID-19 pandemic. PSU Research Review, 8(1), pp.211-226.

[6] Shafiezad, O. and Mostofi, H., 2024. Sentiment analysis of Berlin tourists' food quality perception through artificial intelligence. Tourism and Hospitality, 5(4), pp.1396-1417.Drossos, D., Coursaris, C. and Kagiouli, E., 2024. Social media marketing content strategy: A comprehensive framework and empirically supported guidelines for brand posts on Facebook pages. Journal of Consumer Behaviour, 23(3), pp.1175-1192.

[7] Okeke, N.I., Alabi, O.A., Igwe, A.N., Ofodile, O.C. and Ewim, C.P.M., 2024. Customer journey mapping framework for SMEs: Enhancing customer satisfaction and business growth. World Journal of Advanced Research and Reviews, 24(1), pp.2004-2018.

[8] Inavolu, S.M., 2024. Exploring AI-driven customer service: Evolution, architectures, opportunities, challenges and future directions. International Journal of Engineering and Advanced Technology, 13(3), pp.156-163.

[9] Rawat, A., Kumar, S. and Samant, S.S., 2024. Hate speech detection in social media: Techniques, recent trends, and future challenges. Wiley Interdisciplinary Reviews: Computational Statistics, 16(2), p.e1648.

[10] Arslan, Y., 2025. Online Reputation Management and User-Generated Content. Online Retail Marketing Practice: Technology, Operations, and Globalization, p.115.

[11] Semenda, O., Sokolova, Y., Korovina, O., Bratko, O. and Polishchuk, I., 2024. Using social media analysis to improve E-commerce marketing strategies. International Review of Management and Marketing, 14(4), p.61.

[12] Istanbulluoglu, D. and Sakman, E., 2024. Successful complaint handling on social media predicts increased repurchase intention: The roles of trust in company and propensity to trust. European Management Journal, 42(1), pp.11-22.

[13] Saheb, T., Sidaoui, M. and Schmarzo, B., 2024. Convergence of artificial intelligence with social media: A bibliometric & qualitative analysis. Telematics and Informatics Reports, 14, p.100146.

[14] Alsemaree, O., Alam, A.S., Gill, S.S. and Uhlig, S., 2024. Sentiment analysis of Arabic social media texts: A machine learning approach to deciphering customer perceptions. Heliyon, 10(9).

[15] Antczak, B., 2024. The influence of digital marketing and social media marketing on consumer buying behavior. Journal of Modern Science, 56(2), pp.310-335.

[16] Putri, L.W.B. and Setiawan, B.L.T., 2025. Analyzing the strategic contribution of social media influencers to e-commerce marketing effectiveness. JUMDER: Jurnal Bisnis Digital dan Ekonomi Kreatif, 1(2), pp.12-21.

[17] Odionu, C.S., Bristol-Alagbariya, B. and Okon, R., 2024. Big data analytics for customer relationship management: Enhancing engagement and retention strategies. International Journal of Scholarly Research in Science and Technology, 5(2), pp.050-067.

[18] Pinheiro, J.M.H., de Oliveira, S.V.B., Silva, T.H.S., Saraiva, P.A.R., de Souza, E.F., Godoy, R.V., Ambrosio, L.A. and Becker, M., 2025. The impact of feature scaling in machine learning: Effects on regression and classification tasks. IEEE Access, 13, pp.199903-199931.

[19] Xie, W., Wang, L., Yu, K., Shi, T. and Li, W., 2023. Improved multi-layer binary firefly algorithm for optimizing feature selection and classification of microarray data. Biomedical Signal Processing and Control, 79, p.104080.

[20] Wang, S., Ren, P., Takyi-Aninakwa, P., Jin, S. and Fernandez, C., 2022. A critical review of improved deep convolutional neural networks for multi-timescale state prediction of lithium-ion batteries. Energies, 15(14), p.5053.

[21] Altınel, D., 2025. Development of Deep Learning Optimizers: Approaches, Concepts, and Update Rules. arXiv preprint arXiv:2509.18396.

[22] Givisis, I., Kalatzis, D., Christakis, C. and Kiouvrekis, Y., 2025. Comparing explainable AI models: SHAP, LIME, and their role in electric field strength prediction over urban areas. Electronics, 14(23), p.4766.

[23] https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset?select=sentimentdataset.csv

[24] Gothane, S., Reddy, G.V., Kumar, K.P., Baswaraj, D., Devi, G.P., Thanugundala, S. and Changala, R., 2024. Sentiment analysis in social media using deep learning techniques. Int J Intell Syst Appl Eng, 12(3), pp.1588-1597.

[25] Nurlanuly, A.T. (2025) 'Sentiment analysis of texts from social networks based on machine learning methods for monitoring public sentiment,' ArXiv.org [Preprint].

[26] Rahamana, S.M.U. and Sudhe, S. (2025) 'Sentiment analysis in social media platforms using deep learning and data mining for business intelligence,' International Journal of Research Publication and Reviews, 6(9), pp. 4534–4541.

Design of a Mobile Application with a Hybrid Recommendation System for Connecting Andean Farmers and Urban Consumers in Peru

Jorge Mayta, Astrid Boronda, Juan Montes
Faculty of Engineering, Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Peru

***Abstract*—The dependence on intermediaries within agricultural marketing chains not only reduces producers' profits but also generates consumer distrust regarding the quality and traceability of the products they obtain. This work proposes a mobile application with a hybrid recommendation system aimed at connecting farmers and consumers in Peru. The combination of collaborative filtering and content-based filtering achieves personalization of the shopping experience based on the producer's offer and the user's interests. Likewise, mobile geolocation technology is incorporated to support delivery tracking, together with an offline-first approach to support the application's operation in rural areas with limited connectivity. A preliminary evaluation of the recommendation system, using the Amazon Reviews 2023 dataset (100,000 interactions), compares the hybrid model against collaborative filtering, content-based filtering, and a popularity baseline across RMSE, F1-score@5, catalog coverage, and diversity. The results show that collaborative filtering provides the strongest performance among the models offering meaningful catalog coverage, while the fixed-weight hybrid configuration evaluated in this study did not outperform it, identifying weight optimization as a priority for future work. The offline-first architecture maintains a 100% synchronization rate even under intermittent connectivity conditions, providing preliminary technical evidence under controlled conditions.**

***Keywords*—Andean agriculture; direct trade; hybrid recommendation system; order tracking; mobile application; offline-first**

I. Introduction

Currently, in Peru, there is a structural disconnection between Andean farmers and urban consumers, which negatively impacts both farmers' profitability and end consumers' trust regarding the origin and quality of products. On one hand, farmers depend largely on intermediaries to market their products. In 2019, 46.8% of producers accessed regional, agro-industrial, or export markets; by 2024, this percentage decreased to 45.7% [1], showing a decline in the overall percentage of producers accessing the market. This limitation affects family farming, which represents 99.8% of the country's agricultural units, of which approximately two-thirds correspond to subsistence family farming [2], a segment characterized by lower access to direct markets and, consequently, greater dependence on intermediaries for selling their production. On the other hand, urban consumers are uncertain about the quality, origin, and traceability of agricultural products, which decreases their trust in traditional purchasing channels [3]. This situation shows why traditional supply chain systems are unable to respond to consumers' growing interest in knowing the origin of what they buy, due to the centrality and vulnerability of the information they handle, reducing trust among the actors in the logistics process [4].

This study aims to design, implement, and preliminarily evaluate a mobile application that connects farmers and consumers for the sale of agricultural products through a personalized recommendation system. This will allow farmers to access a direct marketing channel, while consumers will have greater trust and transparency regarding the products.

The main problem lies in the disconnection between Andean farmers and urban consumers, which limits access to detailed product information. Simple approaches, such as product lists, fail because they ignore user preferences and the variation in supply when relying on historical interaction data processing; as a result, they struggle to capture dynamic changes in user interests and product attributes [5]. Furthermore, when there is a lack of initial data, the problem known as "cold start" arises, which affects the performance and accuracy of the recommendation system, becoming evident as the number of new users and items increases [6].

There are several technological solutions aimed at improving agricultural marketing, such as e-commerce platforms, digital agriculture systems, and mobile applications [7]. First, several platforms are not designed for low-connectivity contexts, given that internet use in rural areas in 2024 was 62.6% compared to 86.1% in urban areas, a difference of 23.5 percentage points [8], showing the difficulty of their adoption in Andean rural areas [9]. Second, existing solutions tend to focus only on the transaction itself, that is, buying and selling, without integrating traceability mechanisms [10]. Furthermore, although some systems incorporate recommendations, these are generally basic or not adapted to the agri-food context [11]. Finally, many proposals fail to adequately account for the digital literacy limitations of the agricultural population, which creates barriers to the use of advanced technologies and results in low technology adoption [12]. Likewise, insufficient technical training and the persistence of the digital divide restrict access to learning opportunities and technological resources, representing factors that limit the development and modernization of the agricultural sector [13].

The key components of this research include the use of a hybrid recommendation system that combines collaborative filtering and content-based filtering to improve the recommendations of products offered to users. Likewise, mobile geolocation technology is integrated to support order tracking of agricultural products. The system also integrates a data management module that stores and processes information related to products, users, and transactions, facilitating interaction between stakeholders through an intuitive interface. As a limitation, hybrid recommendation systems may require greater computational complexity and proper integration of multiple data sources, since these can affect their performance.

The main contributions of this work are as follows:

- A mobile application is developed that integrates a hybrid recommendation system to connect Andean farmers with urban consumers in Peru.
- A hybrid recommendation model is proposed that combines collaborative and content-based filtering to improve the personalization of agricultural products.
- An order tracking system based on mobile geolocation is designed and implemented, allowing consumers to monitor the status and route of deliveries.
- A preliminary evaluation of the recommendation system is conducted using a public dataset from the agri-food domain (Amazon Reviews 2023).
- The offline-first architecture is validated through controlled synchronization tests under three network conditions (full connection, total offline, and intermittent connection).
- As future work, the construction of a proprietary dataset from real user interactions and an evaluation with real users (farmers, consumers, and delivery drivers) are proposed.

This article is organized as follows. Section II presents the related work. Section III describes the conceptual background, system architecture, and proposed method. Section IV presents the experimental protocol, results, and discussion. Section V presents the conclusions and future work.

II. Related Work

In recent years, various digital solutions have attempted to connect farmers with consumers to reduce the information gap, increase product visibility, and simplify purchasing decisions. Faced with limitations such as intermediation, the scarcity of structural data in rural environments, and distrust, approaches based on recommendation systems have emerged, with hybrid systems improving the accuracy and adaptability of recommendations in dynamic environments. This section reviews related works that employ this type of approach or similar proposals. These studies show that, although hybrid systems represent an improvement over traditional methods, they still face significant challenges, such as the integration of heterogeneous data and the complexity of managing supply fluctuations.

In [14], the authors propose APGCN-CF, a model based on graph convolutional neural networks that optimizes agricultural product recommendation in rural e-commerce by integrating spatial and temporal information. This approach improves recommendation accuracy, especially for minor products, outperforming conventional methods. However, its performance depends on large volumes of high-quality data and constant connectivity, which limits its applicability in rural environments with restricted technological infrastructure. In contrast, the proposed system adopts a hybrid recommendation approach centered on collaborative filtering combined with content-based filtering, plus a fallback mechanism for cold-start, which seeks to maintain a good level of personalization even when interaction data is limited, without requiring large volumes of high-quality information.

In [15], the author proposes P-FARM, an e-commerce recommendation system based on profit-supported fuzzy association rule mining, which incorporates economic criteria to optimize both recommendation accuracy and transaction profitability. This approach demonstrates improvements in system effectiveness by balancing user satisfaction with business benefits. However, it requires large volumes of consistent transactional data, which restricts its application in informal or low-digitalization agricultural contexts. Furthermore, its emphasis on profit maximization may conflict with equity objectives for producers. By contrast, the proposed approach integrates two distinct components: personalized, consumer-oriented recommendation algorithms and an offline-first architecture that allows farmers and delivery drivers to register and manage their products in areas with limited connectivity, synchronizing once the connection is restored.

In [16], the authors present FarmFresh, a web and mobile application developed in India that allows small farmers to sell their products directly to consumers without intermediaries. Their system incorporates features such as inventory management, order intake, and purchase tracking, which improve the organization and control of sales. The results obtained show an increase in sales and a high level of satisfaction among both farmers and consumers. However, the proposal is oriented toward an environment with better connectivity conditions and does not account for the technological limitations specific to rural areas. By contrast, the proposed system integrates personalized recommendation algorithms and an offline-first architecture oriented toward Andean areas with limited connectivity.

In [17], the authors analyze how the use of digital platforms in Peru influences the profitability of family farmers. The authors find that the use of these tools makes it possible to increase income, reduce transaction costs, and improve efficiency in the marketing of agricultural products. However, they also show low technology adoption due to digital infrastructure limitations and low levels of technological literacy in rural areas. These barriers restrict the impact of existing digital solutions. On the other hand, this work designs a mobile application adapted to these problems through simple interfaces and an offline-first architecture that allows it to function even without a constant connection, facilitating its adoption in the Andean context.

In [12], the authors present AgroTIC, a mobile application developed in Colombia that connects farmers, agronomists, and merchants through the use of technologies such as ICT, IoT, and machine learning, focusing on crop recognition and agricultural information management. This solution seeks to improve productivity and decision-making in the agricultural sector through the use of smart technology. Despite having active users, adoption in rural areas was limited due to a lack of trust in technology and internet connectivity problems. These limitations affect the system's scalability in rural contexts. By contrast, the proposed system focuses mainly on product recommendation, order tracking, and direct trade between Andean farmers and urban consumers, taking intermittent connectivity and ease of use into account from the design stage.

Table I summarizes the key characteristics of the reviewed works compared to the current proposal.

TABLE I. COMPARISON OF RELATED WORK

Work	*Personalized Recommendation*	*Order Tracking*	*Offline-First*	*Direct Trade*
[14] APGCN-CF	✓	✗	✗	✗
[15] P-FARM	✓	✗	✗	✗
[16] FarmFresh	✗	✗	✗	✓
17] Digital platforms (Peru)	✗	✗	✗	✓
[12] AgroTIC	✗	Partial	✗	Partial
Proposal	✓	✓	✓	✓

Beyond merely integrating existing recommendation techniques into a single system, the technical novelty of this work lies in the design of a hybrid model built for the specific constraints of the deployment environment. Instead of relying on external machine learning libraries, both collaborative and content-based filtering are implemented natively in Java 21 using cosine similarity; this keeps the recommendation engine lightweight and free of heavy runtime dependencies that would be difficult to maintain in a resource-constrained, intermittently connected backend. Their outputs are combined through a fixed weighting scheme (45% collaborative, 55% content-based), together with a fallback mechanism that mitigates the cold-start problem for new users and products. As discussed in the Results and Discussion, this fixed ratio was set a priori rather than tuned through systematic search, and the evaluation reported in this study identifies weight optimization as a necessary next step; the contribution of this component lies in the dependency-free, configurable fusion mechanism itself, rather than in a claim that the current weighting is already optimal. This combination of a dependency-free implementation and an explicit cold-start fallback distinguishes the proposed approach from prior agricultural recommender systems, which typically rely on specialized ML infrastructure (e.g., graph neural networks) that assumes larger, denser datasets and more stable computational environments than those available in the target context.

III. CONTRIBUTION AND METHOD

A. Preliminary Concepts

This section presents the main theoretical concepts used in the research. These concepts allow us to understand the use of digital technologies, recommendation algorithms, and supply chain management in the agricultural context, which underpin the development of a mobile application to connect Andean farmers with urban consumers.

1) Agricultural digitalization [18]: Digitalization in agriculture consists of the use of technological tools to optimize agricultural practices and improve production and commercial processes.

In this context, digital platforms, such as mobile applications, make it possible to directly connect farmers with urban consumers, eliminating intermediaries and optimizing logistics.

2) Recommendation algorithms [12]: These are techniques used to predict which items might be of interest to a user based on their previous preferences and behavior patterns.

These algorithms make it possible to personalize user experience, improve customer satisfaction, and optimize sales on agricultural digital platforms.

3) Logistics and supply chain [19]: Logistics comprises the processes of transportation, storage, and distribution of products within a supply chain.

In the agricultural field, efficient supply chain management helps minimize post-harvest losses, ensure product quality, and facilitate traceability, which contributes to building consumer trust.

4) Hybrid recommendation system [20]: A hybrid recommendation system combines more than one recommendation technique with the aim of leveraging the strengths of both systems used and reducing the limitations of each system. It generally integrates methods such as collaborative filtering and content-based filtering to improve the accuracy and relevance of recommendations.

This approach allows the urban consumer to have a personalized shopping experience while, at the same time, increasing the visibility of products offered by Andean farmers within the platform.

5) Collaborative filtering [20]: Collaborative filtering is a recommendation technique that generates suggestions based on the preferences, behaviors, or interactions of users with similar interests. This approach is based on the idea that users who shared consumption patterns in the past may show similar preferences in the future.

This technique makes it possible to recommend agricultural products to a consumer based on the purchases, views, or preferences of other consumers with similar behavior within the platform.

6) Content-based filtering [20]: Content-based filtering is a recommendation technique that uses product characteristics to identify items similar to those the user has previously viewed, purchased, or rated.

In the agricultural context, this approach allows products to be recommended considering attributes such as category, product type, price, origin, or availability.

7) Mobile geolocation [21]: Mobile geolocation is the process by which the approximate geographic location of a mobile device is determined using technologies such as GPS, smartphone sensors, mobile networks, or geofencing mechanisms.

In the proposal, geolocation makes it possible to record the approximate location of delivery drivers or farmers during order transport, facilitating delivery coordination and giving the consumer greater confidence about the status and route of the products ordered.

8) Offline-first architecture [22]: Offline-first architecture is an application design approach that keeps critical functionality available even when there is no internet connection or connectivity is limited. To achieve this, the application prioritizes local storage and allows certain operations to be carried out without permanently depending on the server.

This approach is mainly aimed at farmers and delivery drivers, who may operate in rural areas or routes with intermittent connectivity.

9) Data synchronization in mobile applications [22]: Data synchronization in mobile applications is the process by which information recorded locally on the device is subsequently updated with the central server once the internet connection is restored.

In an offline-first architecture, this process is essential for maintaining data consistency, allowing products registered by farmers and delivery statuses recorded by delivery drivers to be correctly reflected on the platform.

B. System Architecture Overview

The general architecture of the proposal is presented in Fig. 1. The solution was developed as a mobile application in Flutter under an offline-first approach, with a backend implemented in Spring Boot and a central MySQL database. Communication between the mobile application and the backend is carried out through a REST API protected with JWT-based authentication and authorization. As shown in Fig. 1, the application incorporates a local database (SQLite with Drift), an automatic synchronization client, a pending-changes queue, and a geolocation module, allowing consumers, farmers, and delivery drivers to keep using the platform even in scenarios of limited or intermittent connectivity.

On the backend side, business logic is centralized through modules for product and farmer management, orders and deliveries, geolocation, hybrid recommendations, synchronization, authentication, and notifications. The recommendation engine combines collaborative and content-based filtering to generate personalized recommendations, while the cold-start problem is addressed using products with higher ratings, popularity, or demand. Additionally, Azure Storage stores product images, and Firebase Cloud Messaging enables push notifications.

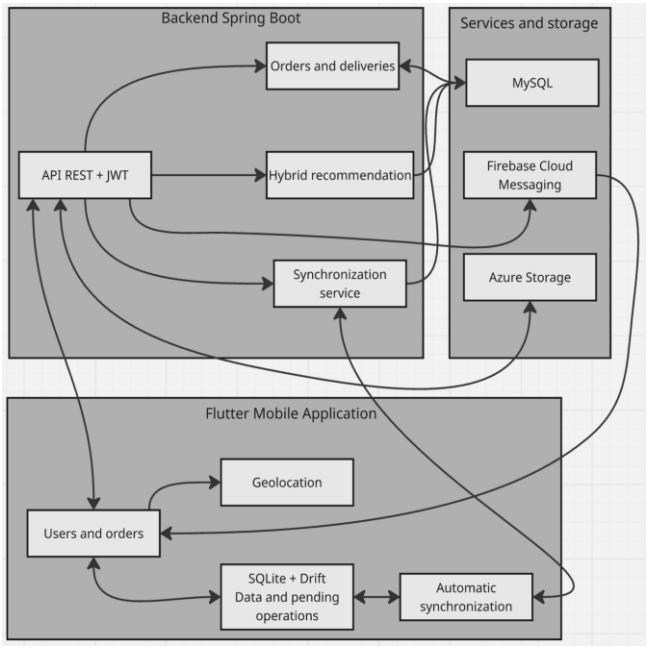


Fig. 1. General architecture of the proposed mobile application.

To protect sensitive location and identity information, the platform's security model addresses a specific threat scenario rather than offering general security guarantees. Payment processing is out of scope, as financial transactions take place outside the application. The primary focus is on safeguarding user identities, maintaining Role-Based Access Control (RBAC) boundaries among farmers, consumers, and delivery personnel, and preserving data integrity during transmission. Authentication and access permissions rely on JSON Web Tokens (JWT), chosen to align with the system's "offline-first" operational capabilities.

Since mobile devices often operate without constant backend connectivity, maintaining session states on the server would disrupt offline usage; instead, self-contained tokens allow for the local verification of authorization credentials between synchronization intervals. Geolocation data collected for delivery tracking requires explicit consent, as it is processed solely during delivery operations and transmitted via encrypted channels. Access to these tracking records is strictly restricted to the relevant consumer, producer, and driver. Furthermore, cached tokens and synchronization queues are kept in encrypted local storage, while location records are deleted once deliveries are completed and any potential disputes resolved. These measures mitigate key risks such as unauthorized access, session hijacking under fluctuating network conditions, and the misuse of location data; however, as they do not constitute a comprehensive security guarantee, the implementation of additional measures, such as formal security audits, is reserved for future stages of the project.

C. Method

The proposal is structured around three main contributions that address the identified problems: personalization of agricultural products through a hybrid recommendation system, logistical coordination through mobile geolocation, and operational continuity through an offline-first architecture for farmers and delivery drivers. These contributions help explain how the solution's main components interact and how technical problems such as cold-start, limited connectivity, and data synchronization are addressed.

The first contribution of this research consists of the implementation of a hybrid recommendation system aimed at improving the shopping experience within the TinkiyAgro platform. This system seeks to facilitate the connection between Andean farmers and urban consumers by generating personalized agricultural product recommendations.

The recommendation module implements a weighted hybrid strategy that combines collaborative filtering and content-based filtering. User-item interactions are built from three behavioral signals: favorites, product reviews, and purchases. Favorites receive a weight of 5.0, reviews use the explicit rating value, and purchases receive a weight of 3.0 plus a quantity-based increment capped at 2.0. These signals are aggregated by user and product.

The collaborative component computes user-user similarity using cosine similarity over the weighted interaction vectors. Candidate products are scored by accumulating the interaction weights of similar users, weighted by their similarity to the target user. The resulting scores are normalized by the maximum score observed in the candidate set.

The content-based component builds a user profile from the attributes of previously interacted products. The implemented features are product category, farmer identifier, sale format, and product popularity. The content score is computed as a weighted combination of category affinity, farmer affinity, sale-format affinity, and popularity, using weights of 0.55, 0.20, 0.15, and 0.10, respectively.

The final hybrid score is computed as 0.45 times the collaborative score plus 0.55 times the content-based score. Products already seen by the user receive a penalty of 0.20 in order to prioritize unseen items. For cold-start users without interaction history, the system falls back to a popularity score based on average rating and number of reviews.

Likewise, the system addresses the cold-start problem, which occurs when a new user does not yet have enough interaction history to generate personalized recommendations. For this case, the platform proposes an initial strategy based on products with higher ratings, popularity, or demand within the system. Subsequently, as the user performs searches, views products, or makes purchases, the system progressively updates its recommendations according to their actual behavior.

As illustrated in Fig. 2, the inclusion of this recommendation system makes it possible to personalize the browsing experience within the platform, facilitate the discovery of new agricultural products, and contribute to increased sales opportunities for registered farmers.

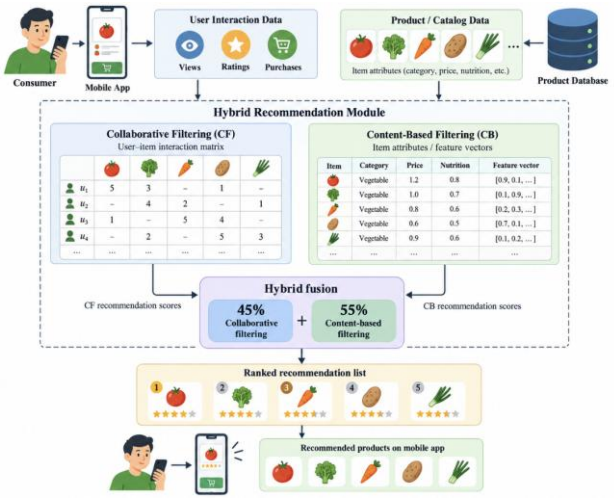


Fig. 2. Operation of the proposed hybrid recommendation system.

The second contribution of this research consists of the implementation of an offline-first architecture aimed at guaranteeing the availability of the application's main functionalities in environments with limited or intermittent connectivity.

As shown in Fig. 3, the system allows information to be temporarily stored on the mobile device, enabling farmers to register products, update them, and view relevant information without needing a permanent internet connection. Similarly, delivery drivers can view assigned orders and record delivery statuses even when there is no network coverage.

Once the device regains connectivity, the locally stored information is automatically synchronized, ensuring data consistency within the platform.

This approach responds to the connectivity conditions present in various Andean rural communities, where limited internet access can affect the continuous use of digital applications. Through this architecture, the goal is to ensure the operational continuity of critical processes such as product publishing and delivery management, reducing dependence on a permanent connection.



Fig. 3. Offline-first architecture for farmers and delivery drivers.

IV. EXPERIMENTS

A. Experimental Protocol

1) Development environment: This work was developed on a computer with a 2.6 GHz 6-core Intel Core i7 processor, an AMD Radeon Pro 560X 4 GB and Intel UHD Graphics 630 graphics, and 16 GB of 2400 MHz DDR4 memory. The mobile application was built using Flutter, and the recommendation engine was implemented in Java 21, running JDK 22.0.2 using the Spring Boot 4.0.1 framework. The application's source code is publicly available: the backend at https://github.com/JuanMontes18/TinkiyAgro and the mobile application (frontend) at https://github.com/TinkiyAgro/TinkiyAgro-Frontend

2) Data used: For the quantitative evaluation of the models, the public Amazon Reviews 2023 dataset (Grocery and Gourmet Food category) [23], published by McAuley Lab, was used. It was selected for its relevance to the agri-food sector and because it contains both explicit (ratings) and implicit (text reviews) interactions between users and products, allowing simulation of the purchasing and rating behavior the system seeks to model. In later stages, a proprietary dataset is planned to be built from real interactions between farmers and consumers within the application.

The evaluation used the Grocery and Gourmet Food category from Amazon Reviews 2023. According to the dataset documentation, this category contains approximately 7.0M users, 603.2K items, and 14.3M ratings. User-item interactions were built from explicit ratings using parent_asin as the item identifier. Items with ratings greater than or equal to 4.0 were treated as relevant for top-k evaluation.

To assess robustness, the experiment was repeated with five random seeds (1, 2, 3, 4, and 5) using an 80/20 user-stratified random holdout split. The same dataset, preprocessing rules, model parameters, and evaluation metrics were kept fixed across runs. Results are reported as mean ± standard deviation.

To validate the offline-first architecture, an automated and controlled test was run on the repositories responsible for local storage (SQLite via Drift) and synchronization with the backend. Network conditions (full connection, total offline, and intermittent connection) were simulated using instrumented remote services, allowing control over network behavior during each test.

B. Result

This subsection details the experiments carried out and the results obtained in each of them:

1) Comparison of recommendation models: Before evaluating the recommendation models, the Amazon Reviews 2023 dataset (Grocery and Gourmet Food category) was processed to prepare an appropriate evaluation subset. Table II outlines the primary attributes of the resulting dataset, detailing the counts of users and items included, total retained interactions, overall sparsity, and the train-test splitting methodology applied consistently across all experimental runs and random seeds.

TABLE II. CHARACTERISTICS OF THE EVALUATION DATASET (AMAZON REVIEWS 2023)

Field	Value
Users	10,459
Items	47,961
Interactions	100,000
Unique user-item pairs	97,884
Sparsity	0.999805
Split	User-stratified random holdout, test_ratio = 0.2
Seeds	1, 2, 3, 4, 5

To evaluate the proposed hybrid system, its performance was compared against a pure collaborative filtering model and a pure content-based model, using the Amazon Reviews 2023 dataset (Grocery and Gourmet Food). The metrics used were RMSE (rating prediction error), F1-score@5 (top-5 ranking quality), Coverage (catalog coverage), and Diversity@5 (intra-list diversity). Table III shows the results obtained for each model.

TABLE III. COMPARISON OF RECOMMENDATION MODELS ACCORDING TO ACCURACY, RANKING, COVERAGE, AND DIVERSITY METRICS

Model	*RMSE*	*F1-score@5*	*Coverage*	*Diversity@5*
Popularity	**2.3199 ± 0.0039**	**0.0245 ± 0.0005**	0.02% ± 0.00%	0.7346 ± 0.0260
Collaborative	2.8651 ± 0.0063	0.0200 ± 0.0005	**11.63% ± 0.15%**	**0.8986 ± 0.0040**
Content Based	2.8881 ± 0.0043	0.0149 ± 0.0002	5.47% ± 0.07%	0.3952 ± 0.0088
Hybrid	3.0342 ± 0.0034	0.0185 ± 0.0005	10.39% ± 0.11%	0.6072 ± 0.0042

Table III shows that the popularity baseline achieves the lowest RMSE (2.3199) and the highest F1-score@5 (0.0245), reflecting a well-documented behavior of top-k ranking metrics: recommending the most frequently purchased items maximizes short-term accuracy and ranking overlap with a sparse test set, but at the cost of catalog coverage, which remains close to zero (0.02%), effectively recommending the same narrow set of products to nearly all users, which runs counter to the platform's goal of increasing visibility for a broader base of farmers and products. Among the models that provide meaningful catalog coverage, collaborative filtering achieved the best result across all four metrics (RMSE 2.8651, F1-score@5 0.0200, Coverage 11.63%, Diversity@5 0.8986), outperforming both content-based filtering and the hybrid configuration evaluated in this study. The hybrid model, combining collaborative and content-based scores with a fixed 45%/55% weighting, did not outperform collaborative filtering alone at this scale. This indicates that the specific weighting ratio used was not optimal for the sparsity and interaction patterns observed in this dataset; the 45/55 split was selected a priori rather than tuned through systematic search. Optimizing this parameter, potentially through per-context rather than fixed weighting, is identified as a priority for future work.

Rather than presenting the fixed-weight hybrid configuration as an optimized final model, the contribution of

this component lies in the native, dependency-light implementation described in Section II, together with a reproducible evaluation protocol, repeated runs with standard deviations, and a popularity baseline that makes the current limitation transparent: at the tested weighting ratio, the hybrid model does not outperform collaborative filtering alone. Reporting this result, rather than obscuring it, identifies the necessary next step for this line of work: a systematic search over the weighting parameter, rather than assuming the current fixed ratio is production-ready.

2) Validation of offline-first functionality: To evaluate the system, 4 key flows were analyzed: product creation, product editing, pickup point creation, and marking an order as ready. Each one was evaluated under 3 different network conditions. The study covered 240 operations, running 20 per flow in each scenario evaluated. The integration of local logging was validated through the syncOperations table in Drift/SQLite, while synchronization effectiveness was determined by checking whether operations changed to the synced state. A Stopwatch was also used to quantify latency, measuring the interval from the start of the process to its successful confirmation. Table IV presents the results obtained.

TABLE IV. RESULTS OF COMPLETENESS, SYNCHRONIZATION, AND LATENCY OF THE OFFLINE-FIRST ARCHITECTURE

Network Condition	*Operation*	*Complete local registration*	*Successful synchronization after reconnection*	*Synchronization latency*
Full connection	80	100%	100%	0.031 s
Total offline	80	100%	100%	0.049 s
Intermittent connection	80	100%	100%	0.311 s

The "total offline" scenario does not imply immediate synchronization with the server while offline. Instead, it means that all operations were saved in the local SQLite queue during the outage and successfully synced once the connection was restored. Thus, the recorded latency reflects only the time taken by the synchronization process after triggering the routine for pending tasks.

C. Discussion

1) Comparison of recommendation models: The hybrid model should not be interpreted as the best model across all individual metrics. Instead, the results show different trade-offs. The popularity baseline achieved the best RMSE and F1@5 in this highly sparse sample, which indicates that frequently rated and highly rated products are easier to recover under limited interaction density. Collaborative filtering achieved the highest coverage and Diversity@5, suggesting broader catalog exploration when enough overlapping user behavior exists. The hybrid model combines collaborative and content signals, but in this configuration it should be presented as a balanced alternative rather than as a universal improvement over all baselines.

2) Validation of offline-first functionality: As shown in Table IV, the offline-first architecture guarantees 100% effectiveness in local registration and synchronization under all three conditions analyzed, ensuring the integrity of all operations regardless of network stability. However, a notable variation in latency is evident: while in full-connection scenarios the average is 0.031 s, this value rises to 0.311 s under intermittent connections. This increase, close to ten times the initial figure, is attributable to the additional processing required to detect and transmit the queue of pending operations after reconnection.

This result is consistent with the expected behavior of an architecture that seeks to prioritize data integrity, ensuring that nothing is lost, over synchronization speed when the network is unstable, which is especially relevant for the context of Andean rural areas with limited connectivity.

It should be noted that, to support the experiment's repeatability, instrumented services were used to simulate network conditions. Consequently, the findings reflect how the offline-first architecture behaves in controlled environments, which means they are not a direct measure of the inherent volatility of mobile networks in real-world scenarios. This distinction represents an important limitation that should be addressed in future validations carried out under uncontrolled network conditions.

Upon reconnecting, the system automatically initiates synchronization to upload all pending local operations to the server. However, the present build lacks mechanisms for resolving concurrent editing conflicts and managing retry attempts for failed transmissions. Handling these edge cases, such as duplicate operations, expired authentication tokens, application restarts, and simultaneous edits, remains a key priority for future work.

V. CONCLUSION

TinkiyAgro was developed as a mobile application that connects Andean farmers, consumers, and delivery drivers through a hybrid recommendations system, a GPS/GNSS-based delivery-tracking module, and an offline-first data architecture. From the outset, the recommendation module includes a fallback strategy to mitigate the cold-start problem for users without purchase history and adapts suggestions progressively as interaction data becomes available.

The evaluation focused on two of these components under controlled conditions: the recommendation model, assessed through RMSE, F1-score@5, Coverage, and Diversity@5 on the Amazon Reviews 2023 dataset, and the offline-first architecture, assessed through local-registration completeness, synchronization success rate, and latency under simulated network conditions. The results provide preliminary technical evidence that both components operate as intended under these controlled scenarios. The geolocation-based delivery-tracking module was implemented but not subjected to a comparable empirical evaluation; its accuracy, update latency, and behavior under connectivity loss remain untested. Validation with real users (farmers, consumers, and delivery drivers), including usability, perceived trust, and field performance of the tracking

module, remains pending and is identified as a priority for future work. A geolocation-based delivery tracking module was implemented but did not undergo a comparable empirical evaluation; details such as its accuracy, update latency, and behavior during connectivity loss remain untested. Validation with real users (farmers, consumers, and delivery personnel), covering the module's usability, perceived trustworthiness, and field performance, is still pending and has been identified as a priority for future work.

In conclusion, the results obtained provide preliminary technical evidence for two core components of the proposed solution: a recommendation module comparing hybrid, collaborative, and content-based filtering under a reproducible, multi-seed evaluation protocol, which found that collaborative filtering alone provided the strongest results among the evaluated models and identified weight optimization as a necessary next step for the hybrid formulation to realize its intended benefit, and an offline-first architecture designed to support operational continuity under limited or intermittent connectivity. These findings should be interpreted as evidence of technical feasibility under controlled conditions rather than as validation within the Andean rural context itself. As future work, it is proposed to evaluate the complete application, including the geolocation module, with real users in field conditions, to assess usability, scalability, and the system's actual impact on farmer income and consumer trust.

DECLARATIONS

A. Declaration on Generative AI Use

During the preparation of this study, the authors used a generative AI tool to help refine the language and formatting consistency, as well as to generate illustrative figures (Fig. 2 and Fig. 3). All AI-assisted text and figures were critically reviewed, verified, and edited by the authors, who assume full responsibility for the accuracy and content of the final paper. The generative AI tool played no role in the development of the scientific ideas, the design of the recommendation algorithm, the experimental methodology, or the interpretation of the results.

B. Conflict of Interest

The authors declare no conflict of interest.

C. Data Availability

The Amazon Reviews 2023 dataset used for the recommendation model evaluation is publicly available at https://amazon-reviews-2023.github.io/.

REFERENCES

[1] Instituto Nacional de Estadística e Informática (INEI) and Ministerio de Desarrollo Agrario y Riego (MIDAGRI), "Productores agropecuarios: Principales resultados de la Encuesta Nacional Agropecuaria (ENA), 2018–2019 y 2022–2024," INEI, Lima, Peru, 2025. [Online]. Available: https://www.gob.pe/institucion/inei/informes-publicaciones/6879473-productores-agropecuarios-principales-resultados-de-la-encuesta-%20nacional-agropecuaria-ena-2018-2019-y-2022-2024

[2] C. E. Cabrera Cevallos and A. P. De la O Campos, "La agricultura familiar en el Perú: Brechas, retos y oportunidades," Economía del Desarrollo Agrícola de la FAO, Estudio Técnico No. 28, FAO, Rome, Italy, 2023, doi: 10.4060/cc4897es.

[3] C. Duong, T. Dao, T. Vu, T. Ngo, and M. Nguyen, "Blockchain-enabled food traceability system and consumers' organic food consumption: A moderated mediation model of blockchain knowledge and trust in the organic food chain," Sustainable Futures, vol. 6, Art. no. 100316, 2024, doi: 10.1016/j.sftr.2024.100316.

[4] G. Ji, S. Zhou, K.-H. Lai, K. H. Tan, and A. Kumar, "Timing of blockchain adoption in a supply chain with competing manufacturers," International Journal of Production Economics, vol. 247, Art. no. 108430, 2022, doi: 10.1016/j.ijpe.2022.108430.

[5] Z. Hou, F. Bu, Y. Zhou, L. Bu, Q. Ma, Y. Wang, H. Zhai, and Z. Han, "DyCARS: A Dynamic Context-Aware Recommendation System," Mathematical Biosciences and Engineering, vol. 21, no. 3, pp. 3563–3593, 2024, doi: 10.3934/mbe.2024157

[6] H. Yuan and A. Hernandez, "User Cold Start Problem in Recommendation Systems: A Systematic Review," IEEE Access, vol. 11, pp. 136958–136977, 2023, doi: 10.1109/ACCESS.2023.3338705.

[7] P. Devika, G. Mathu Kumar, P. Nagul Kumar, and S. Naveen Prabhu, "Agriconnect: A mobile solution for direct farmer-to-market trade," in Proceedings of the International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024), Atlantis Press, 2025, pp. 1413–1419, doi: 10.2991/978-94-6463-718-2_117.

[8] A. Ramírez García, "Sub-national statistical measurement of universal and meaningful connectivity in Peru," International Telecommunication Union (ITU), Project UMC, Dec. 2025. [Online]. Available: https://www.itu.int/itu-d/sites/projectumc/wp-content/uploads/sites/42/2025/12/rev-SUB-NATIONAL-STATISTICAL-MEASUREMENT-OF-UNIVERSAL-AND-MEANINGFUL-CONNECTIVITY-IN-PERU.pdf

[9] D. J. Choruma, T. L. Dirwai, M. J. Mutenje, M. Mustafa, V. G. P. Chimonyo, I. Jacobs-Mata, and T. Mabhaudhi, "Digitalisation in agriculture: A scoping review of technologies in practice, challenges, and opportunities for smallholder farmers in Sub-Saharan Africa," Journal of Agriculture and Food Research, vol. 18, Art. no. 101286, 2024, doi: 10.1016/j.jafr.2024.101286.

[10] M. Reitano, G. Pappalardo, R. Selvaggi, C. Zarbà, and G. Chinnici, "Factors influencing consumer perceptions of food tracked with blockchain technology: A systematic literature review," Applied Food Research, vol. 4, no. 2, Art. no. 100455, 2024, doi: 10.1016/j.afres.2024.100455.

[11] F. ZhuanSun, J. Chen, W. Chen, Y. Sun, and W. Hang, "Analysis of Precision Service of Agricultural Product e-Commerce Based on Multimodal Collaborative Filtering Algorithm," Mathematical Problems in Engineering, vol. 2022, Art. no. 8323467, 2022, doi: 10.1155/2022/8323467.

[12] C. Hinojosa, K. Sánchez, A. Camacho, and H. Arguello, "AgroTIC: Bridging the gap between farmers, agronomists, and merchants through smartphones and machine learning," arXiv preprint arXiv:2305.12418, 2023, doi: 10.48550/arXiv.2305.12418.

[13] D. B. Escudero Yupanqui and W. Carmona Núñez, "Análisis de los desafíos y oportunidades que ofrece la implementación del Internet de las Cosas (IoT) en el marco de la Agricultura 4.0 en el Perú," Ciencias e Ingeniería, vol. I, no. 1, pp. 28–54, Apr. 2025. [Online]. Available: https://ctscafe.pe/index.php/cienciaingenieria/article/view/365

[14] W. Hou, K. Zhu, P. Liu, H. Jiang, Y. Zhang, and X. Yu, "APGCN-CF: A Spatio-Temporal-Aware Graph Convolutional Network Framework for Minor Agricultural Product Recommendation in Rural E-Commerce," Applied Sciences, vol. 15, no. 8, Art. no. 4362, 2025, doi: 10.3390/app15084362.

[15] O. Dogan, "A recommendation system in e-commerce with profit-support fuzzy association rule mining (P-FARM)," Journal of Theoretical and Applied Electronic Commerce Research, vol. 18, no. 1, pp. 43–61, 2023, doi: 10.3390/jtaer18020043.

[16] E. Archana, M. T. Fathima, B. Roshini, and A. Roshal, "FarmFresh: A Mobile/Web App for Connecting Farmers and End Users," TIJER – International Research Journal, vol. 11, no. 2, pp. 622–626, 2024. [Online]. Available: https://tijer.org/tijer/viewpaperforall.php?paper=TIJERB001098

[17] L. Arteaga and E. Villarroel, "Plataformas digitales y rentabilidad de agricultores del valle del Mantaro, Huancayo," Quipukamayoc, vol. 31, no. 66, pp. 33–44, 2023, doi: 10.15381/quipu.v31i66.26270.

[18] A. Gabriel and M. Gandorfer, "Adoption of digital technologies in agriculture—An inventory in a European small-scale farming region," Precision Agriculture, vol. 24, pp. 68–91, 2023, doi: 10.1007/s11119-022-09931-1.

[19] M. Krstić, G. P. Agnusdei, S. Tadić, and P. P. Miglietta, "Prioritization of e-traceability drivers in the agri-food supply chains," Agricultural and Food Economics, vol. 11, Art. no. 42, 2023, doi: 10.1186/s40100-023-00284-5.

[20] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: A survey," IEEE Computational Intelligence Magazine, vol. 19, no. 2, pp. 78–95, 2024, doi: 10.1109/MCI.2024.3363984.

[21] M. Emish, Z. Kelani, M. Hassani, and S. D. Young, "A mobile health application using geolocation for behavioral activity tracking," Sensors, vol. 23, no. 18, Art. no. 7917, 2023, doi: 10.3390/s23187917.

[22] C. Escobar-Velásquez, A. Mazuera-Rozo, C. Bedoya, M. Osorio-Riaño, M. Linares-Vásquez, and G. Bavota, "Studying eventual connectivity issues in Android apps," Empirical Software Engineering, vol. 27, Art. no. 22, 2022, doi: 10.1007/s10664-021-10020-6.

[23] McAuley Lab, "Amazon Reviews 2023," Hugging Face Datasets, 2023. [Online]. Available: https://amazon-reviews-2023.github.io/

AI-Driven Personalized Learning Framework for Computing Fields Recommendations in Secondary and Pre-University Contexts

Anas Inuwa Usman[1], Shahida Sulaiman[2], Ruhaidah Samsudin[3]
Faculty of Computing, Universiti Teknologi Malaysia, 81310, Johor Bahru, Johor, Malaysia[1, 3]
Knowledge Wonders PLT, 81930, Bandar Penawar, Johor, Malaysia[2]

***Abstract*—Students or learners, especially those at the secondary and pre-university levels, face challenges in identifying suitable computing fields for personalized educational resources towards their tertiary education specialization pathways. Traditional advising methods/approaches such as guidance from school counselors and teachers often lack personalization and practical relevance because their knowledge of the field is limited. This research aims to close this gap by proposing Computing Field Recommender (ComReF), an end-to-end Machine Learning framework that integrates student assessment and a profile built from demographic attributes and assessment performance, with semantic resource generation to deliver personalized, interpretable, and pedagogically grounded recommendations. A Random Forest-based multi-class classification strategy predicts suitable computing fields and corresponding proficiency levels, with empirical results showing improved performance over Decision Tree, Logistic Regression, and Naïve Bayes classifiers. NLP was also used for string cleaning and to automatically generate URLs for resources such as YouTube, Wikipedia, and Coursera. Consequently, all generated resources undergo a domain-informed validation protocol based on industry and academic checklists/benchmarks, ensuring their relevance and quality. The evaluation was conducted on a sample of 500 labeled instances using ChatGPT and available assessment data. The results show a strong classification accuracy (F1-score ≥ 0.8 for a well-represented field) in skill-level classification and field-level predictions across the computing fields/subfields that indicate moderate accuracy (with a weighted F1 = 0.59), reflecting real-world data imbalance, and producing relevant resource sets that are topically consistent with the predicted computing specializations and career goals. By bridging predictive modeling with semantic web navigation, ComReF offers a suitable solution for computing field recommendations and helps reduce gaps in traditional advising approaches.**

***Keywords*—Artificial intelligence; machine learning models; pre-university level; personalized learning; computing fields; recommender systems**

I. Introduction

The choice of specializations in Computing, particularly among secondary and pre-university students, depends on several factors. These factors include financial barriers, limited knowledge of the nature of various fields and their requirements, tuition fees for certain programs, the educational institutions offering the programs, and concerns about the ability to meet academic requirements [1]. Recommender systems provide evidence-based information to help learners select courses by suggesting alternatives based on their interests and performance profiles and aligning with their academic goals and objectives [2]. In addition, these systems contribute to helping students meet curricular needs, expand their career opportunities, and engage in intellectual growth [3].

Advancements in artificial intelligence (AI) have allowed recommender systems to streamline individualized recommendations [4]. However, the extent to which AI-based systems can influence students' field choices remains under-researched. It is necessary to gain more understanding of how students perceive and expect such systems [5]. In addition, AI chatbots have become a topic of interest among scholars worldwide and have the potential to change educational systems in various ways. They can offer real-time support, answer questions, explain concepts, and provide individualized learning materials. Moreover, chatbots can serve as an additional tool of virtual teaching [6].

Over the years, students have often struggled to choose elective courses because of the growing number of choices available to them. As elective options expand, course recommender systems have become more relevant for accurate choices [7]. The significant increase in AI as a digital assistant is especially impressive, as it provides individualized learning by incorporating educational assets based on students' personal preferences and the requirements of their discipline [8]. They support teachers and learners across various dimensions of the educational process [9]. The potential to address the limitations of conventional career-support mechanisms is significant in the context of integrating AI-driven conversational frameworks into career guidance platforms [10]. Research suggests that AI-driven advising or career guidance can be effectively operationalized when supported by strong technological foundations, such as advanced natural language processing frameworks and secure, cloud-based machine learning services that provide effective, personalized guidance [11]. In addition, effective systems should be flexible enough to adapt to new work-market factors and institutional developments [12].

On the other hand, even with the rising demand for professionals holding bachelor's degrees in Computing-related fields, undergraduate enrollment in Computer science and Computer engineering programs has shown a marked decline in recent years [13]. An earlier alarm was also raised by Lent

[14], [15]. The continuing inadequate representation of women and minority groups in the computing sector, together with an associated downward trend in the number of students seeking computing courses, continues to raise a red flag despite the rising labor market demand [16].

The issue highlights the need to understand psychosocial and contextual factors that influence students' choices about whether to pursue or avoid computing disciplines, especially among underrepresented populations. In a larger sense, it reflects efforts to increase the number of students recruited and retained in STEM fields [17]. The dwindling interest in computer-related programs is a long-standing international issue with major long-term implications. This tendency underlines the dire need to come up with new methods of popularizing those fields and to reconsider the avenues through which they are marketed [18] Cohon et al. (2018). Student representatives of different groups still consider the interest in the career field as one of the most powerful factors in career decision-making [19].

Although there are attempts to democratize access to Computer Science, the field remains characterized by significant skewness beyond existing gender and socioeconomic imbalances. The level of computing interest and engagement is comparatively stronger and higher among male students, and undergraduates are prone to leaving school, especially among low-income earners [20], [16]. To this end, it is necessary to provide extensive, impartial exposure to computing, an indispensable requirement when increasing the level of involvement and broadening opportunities in these fields [21]. Computer Science is a crucial field in developing future technological leaders; however, the accessibility and inclusion problems are often ignored in existing curricula [22]. There is a growing need for practical guidance on accessible computing that equips learners with strategies to design more inclusive learning environments. This guidance must clarify the ethical, legal, and pedagogical requirements of accessibility and how inclusive practices can bring benefits to all learners, including learners with disabilities [23]. In addition, there is a need to differentiate teaching accessibility as a discipline on its own and the integration of accessibility principles into instructional practice as an educator [24].

Computing has been encompassed in a wide field of study known as Computer Science; it has since split into various specialized fields, including AI, cybersecurity, software engineering, network engineering, business intelligence, sustainable computing, and data science [25]. Each of these fields requires distinct competencies, offers unique career pathways, and follows differentiated learning trajectories [26]. The diversification can be seen as a sign of the maturation of the discipline and its embedding into almost all sectors of the world economy [18]; [27]. As a result, it poses a great cognitive and navigational dilemma to the students who, in most cases, must make high-stakes specialization choices without enough exposure, experience, or guidance [28].

Academic advising, mostly based on stagnant curricula or general career presentations, cannot even keep up with the fast-changing environment and does not fulfill the dynamic counseling requirements of the students [29], [30]. Conversely, recommender systems are also used on online learning platforms like Coursera and edX, where they generally become active after enrolment and provide recommendations based on learners' previous interactions. Nonetheless, these systems rarely offer recommendations for choosing a field of study based on diagnostic data or assessment information [31]. Similarly, most educational data mining methods focus on predicting academic achievement or the risk of dropping out of school, rather than providing an all-inclusive, easy-to-use learning-to-career map for learners [32].

There is still a notable gap in recommender systems capable of linking diagnostic assessments with the automated generation of semantically relevant learning resources [33]. Most AI-driven tools provide only high-level labels, such as “AI,” or numerical competency scores, leaving students to locate their own learning materials. This process is often time-consuming and susceptible to information overload [34]. Even when learning resources are recommended, they are seldom evaluated against established pedagogical standards or current industry benchmarks [35], [36]. In the context of our research, these three important terms are thus defined as below:

Semantic resource generation is the process of creating platform-specific URLs with full canonical field names (e.g., Business Intelligence) instead of short codes or initials (“BI” for Business Intelligence and “AI” for Artificial Intelligence), which overcomes problems of fields that may have the same initials as Cryptography and Cybersecurity. It also ensures lexical matching in search interfaces like Coursera and Wikipedia.

Interpretable: implies that it is possible to trace predictions to input features by using model-internal measures of feature importance and explicit, auditable code-to-name mapping, which removes the need to use black-box components like deep learning models or large language models (LLMs).

Actionable and coherent: refers to an output that consists of a non-redundant collection of screened links, each of which retrieves topic-aligned resources with explicit descriptors of skills, such as in React.js, Data Analysis, and Ethical Standards.

Existing studies have focused on course selection [2], [5], career guidance [10], learner profiling [37], and learning paths [38]. However, these functions are often addressed separately, with limited integration to guide secondary and pre-university learners toward suitable computing fields. To address this gap, ComReF integrates learner profiles and assessment data to recommend computing fields, identify skill levels, and provide relevant learning resources. Its main contribution is a lightweight and interpretable framework that supports early computing specialization and complements teacher and academic-adviser guidance.

To address the limitations mentioned earlier, ComReF is proposed as an end-to-end machine learning framework that transforms student assessments and profiles (including demographic and preference information) into personalized learning recommendations. It aims to guide and expose students to suitable computing fields from the secondary level based on their profiles and cognitive skills.

Hence, ComReF aims to reduce the gaps in the following aspects:

Learner Profiling: Learning unfolds differently for every individual. With the emergence of "Internet+" education, personalized learning has gained significant [39]. Digital learning materials have led to a significant rise in online learners in many aspects of activities [40]. As such, learners often face the problem of information overload and feel lost during the learning process [41]. These limitations may create a mismatch between a learner's actual competence and the difficulty of the suggested resources, increasing the risk of cognitive overload. Our approach aims to combine the ability to recognize and understand individual needs.

Field Mapping: The abundance of available learning resources has increased manifold, and the methods of resource selection have also changed in tandem with the learners. In that regard, learning materials have grown in significance [42]. Learning pathways are supposed to be adjusted to the educational field resources depending on learners' specific profiles [37]. We need to map resources by field, not randomly.

Resource Generator: A framework should allow searching by the entire field name to make the searches that are discovered pertinent and educational [43]. ComReF provides an organized learning path to help learners progress and optimize the entire learning experience.

Domain-Informed-Validation: This task is important as well as complicated, both in design, development, and evaluation of the learning environment in ways that support the distinct learner needs in a particular domain or field. A learning environment has several dimensions, such as physical, social, psychological, and pedagogical dimensions, that all constitute the environment where learning occurs [44]. In particular, the learning experiences refer to various interactions and different situations that stimulate the perceptions of learners [45]. ComReF screens suggested resources using a checklist based on compatibility with the requirements of the Malaysian education blueprints and industry standards.

Therefore, there is a need to develop accurate student profiles and models that analyze learners' affective, knowledge, personality, and skill levels. Thus, AI solutions offer considerable benefits, as they can incorporate reasoning into instructional decision-making and control potential sources of uncertainty. Therefore, they enable more efficient teaching and learning systems [46]. The following sections will elaborate on related works, the proposed framework, its implementation and prototyping, discussion, and conclusion.

II. Related Works

This section presents existing literature on educational recommender systems, career guidance in Computing education, semantic resource generation and validation, and validation pathways, which provide the foundation for this work and its relevance to targeted learners.

A. Educational Recommender Systems: From Courses to Careers

The evolution in educational recommender systems has shifted from primitive collaborative filtering methods such as course recommendations based on peer activity [47] to advanced hybrid models that combine knowledge tracing, learner profiles, and contextual metadata [48]. For instance, Coursera, edX, and some similar platforms currently adopt deep learning models to personalize course sequences, with studies indicating improved click-through rates when recommendations align with learners' goals [2]. Regardless of such developments, these systems operate reactively, requiring prior engagement [49]. Massive Open Online Courses provide a rich range of learning materials; however, they often overwhelm students due to their enormous volume, which results in problems like data sparsity and cold-start problems [50].

To alleviate these hindrances, EGRec was proposed [49]. It is an innovative course recommendation model that augments recommendation accuracy, diversity, and relevance by combining knowledge graphs and heterogeneous graph attention networks. The model uses multimodal information and dynamically selected subgraphs to improve performance, but it does not focus on field diversification, as seen in the field of Computing. Most educational recommender systems consider a field as a label of a static unit, but not as a semantic unit [51]. Thus, a learner who is designated as fit for AI, such as Generative AI, Computer Vision, or AI Ethics, a field in which entirely different skills are required, is not provided with any additional guidance. A survey of major MOOC platforms shows that only a small number of studies distinguish sub-fields of computing from broad categories, such as Networking, Software Engineering, and Multimedia, and tend to include separate career paths in excessively general categories [52]. This reduced granularity limits practical usefulness, as computing job advertisements are becoming more demanding in more specific areas.

B. Career Guidance in Computing Education

Selecting a computing specialization is a significant challenge [53]. An in-depth examination of how Computer Science and Computational Thinking are taught at lower levels of education is essential to creating consistent, high-quality courses and implementing professional development programs [54]. Computing education has become a high-profile research area in major universities, and institutions, industries, and communities have launched initiatives; however, issues still hinder the widespread implementation of computing education at the K-12 level [55]. Computational Thinking and Problem-solving skills have become important in students' development. Digital and serious games are also entering education at an affordable cost and promise to enhance students' learning [56]. This is comparable to findings of the American National Academies, which warned that without proactive advice, students, especially underrepresented ones, would often gravitate towards familiar or oversaturated fields such as web development, which would overlook high-opportunity ones like DevOps and other emerging specialties [18]. The traditional advising interventions are not very scalable [57]. A randomized controlled trial showed that one-on-one career counseling can enhance field alignment, but it is time-consuming. Moreover, this resource-intensive approach is not feasible for large-scale enrollment within a limited timeframe [58].

C. Semantic Resource Generation and Validation

Recent studies have begun to integrate prediction with outcomes that can benefit learners. Anomalous state of knowledge theory suggests that learners require contextualized information pathways, rather than merely labels [59]. Building on this concept, systems like Path2Learn were realized [60]. The focus on personalized learning paths has become increasingly prominent in education. Tran and team [38] examine the performance evaluation of user-based Collaborative Filtering and Content-based recommendation techniques to develop innovative RSs specifically designed for Information Systems (IS) students. By integrating a primary dataset grounded in the students' knowledge-skill-attitude framework, the study addresses the challenge of navigating the extensive range of learning options in IS. Wan and Lei [61] outline the undisputed benefits of accessible online resources; however, the abundance of these resources often leads to confusion and decision paralysis, so individualized, well-designed guidance is also required. On the same note, LLM-based recommender systems can rank coherent resource lists, but they are vulnerable to hallucination, especially in technical fields [62]. This can result in suggestions for non-existent courses or outdated tools. Validation is often treated as an afterthought in educational AI systems. Though the research of Seo [63] incorporated an expert opinion in the AI sector, which is usually limited to scholars rather than industry practitioners. The advocates of human-in-the-loop AI insist on the necessity of introducing additional validation procedures, but few systems manage to adopt the strategy [64]. This mismatch is alarming because industry expectations often differ from academic programs. For example, DevOps job descriptions often require cloud certifications, but many college programs do not include cloud platform training in their coursework. Therefore, the idea of generating and validating resources remains unexplored.

D. Validated Pathways

Uncovering meaningful hidden patterns in learner data is highly valuable within educational technology. Research on personalized, full-path learning recommendations is especially critical for advancing the development of more sophisticated adaptive learning systems [65]. The surge in online courses makes it harder for learners to select suitable learning paths. Recommender systems help by analyzing past interactions to predict interests and suggest relevant next steps [66]. In summary, none of the related studies focus on full-fledged computing field recommendations based on learners' profiles and assessment scores, using an ML approach to recommend a field alongside learning resources.

For better presentation, studies have explored educational recommendations from different perspectives, including course selection, career guidance, learner profiling, course advising, and personalized learning paths. However, their objectives and recommendation components differ from those integrated in ComReF. To clarify these differences and position the proposed framework's contribution, Table I compares representative related studies by focus and recommendation capabilities.

TABLE I. COMPARISON OF COMREF AND SOME EXISTING STUDIES

Study	Focus	Personalization	Field Guidance	Skill Level	Learning Resources
[5]	Course selection	✓	–	–	–
[10]	Career guidance	✓	–	–	–
[29]	Course advising	✓	–	–	–
[31]	Labor-market-driven education	✓	–	–	✓
[37]	Learner profiling/course recommendation	✓	–	–	✓
[38]	Personalized learning paths	✓	–	–	✓
ComReF	**Computing-field guidance**	✓	✓	✓	✓

Overall, ComReF differs from the reviewed approaches by integrating learner profiling, computing-field recommendation, skill-level identification, and learning resources for secondary and pre-university learners.

III. PROPOSED AI-DRIVEN RECOMMENDATION FRAMEWORK

The proposed ComReF is an end-to-end framework that converts a student's multidimensional aptitude profile (demographic, proficiency, and assessment data) into a personalized learning pathway. The system integrates machine learning models and prediction algorithms, semantic expansion, and cross-platform resource generation (see Fig. 1). It integrates three layers (Data/Interaction Layer, AI Process Layer, and Feedback/Resources/Validation Layer) to deliver personalized learning recommendations.

The Data/Interaction Layer first collects the student's basic information, learning preferences, and performance assessments. These inputs are then used by the AI Process Layer to predict both the optimal field and the skill level of each student using a random forest classifier. It is particularly effective for multi-class prediction tasks, which are essential for recommending diverse computing fields such as Software Engineering, Data Science, and AI. In addition, NLP cleans and normalizes data and generates web search queries to retrieve relevant material. When it generates predictions, the recommender module uses field and skill predictions to deliver personalized resources, including YouTube tutorials, PDF documents, and other pages of interest. These resources are obtained by field, and content curation is adapted for semantic web search. More importantly, all resources are aligned with the industry skills and educational blueprints alignment checklist. ComReF tracks students' interactions with recommended resources and uses the data to make future recommendations.

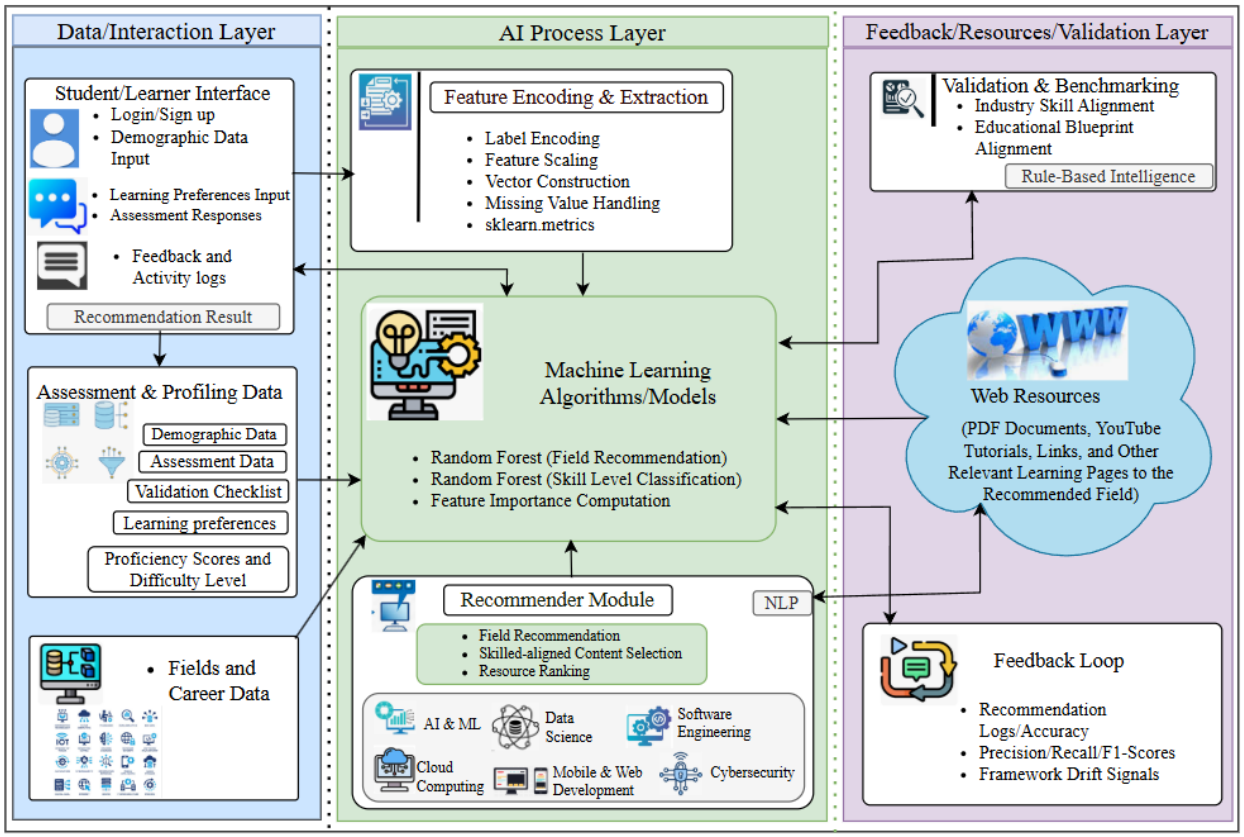


Fig. 1. Computing field recommendation framework (ComReF).

This continuous feedback loop improves accuracy and supports more personalized guidance as students navigate computing resources. The validation layer ensures that only the most relevant and up-to-date learning resources are recommended, ensuring all learners have access to appropriate materials. To improve ComReF's effectiveness, the feedback component incorporates precision, recall, and F1-scores, which are vital for evaluating performance. Drift signals monitor shifts in data patterns over time.

The remaining subsections detail how the framework operates and how the functional stages contribute to the overall recommendation process. Schematic flowcharts and mathematical formulations clarify the workflow, making the underlying reasoning, decision processes, and evaluation procedures clear.

A. Implementation Environment, Tools and Datasets

The implementation environment is a Jupyter Notebook interface for interactive prototyping and documentation, managed through Anaconda (Python version 3.9+). The main libraries deployed are scikit-learn for Random Forest classifiers, pipelines, and encoders; pandas and numpy for data loading, transformation, and analysis; urllib.parse for safe URL encoding across platforms; and re (regular expressions) for text cleaning and query sanitization. The framework does not require GPU support. However, accessing the generated links to external learning resources requires an Internet connection. The dataset requirements include:

Profiling data, consisting of demographic/contextual information, grade/level, learning attitude, motivation, goals, interests, and preferences, while cognitive attributes cover mathematical ability, logical and analytical reasoning, problem-solving, creativity, communication, teamwork, and self-learning readiness.

A field and career goals dataset, comprising identified Computing fields assessment scores mapped to the seven major computing fields: Computer Science, Software Engineering, Information Technology, Information Systems, Computer Engineering, Cybersecurity, and Data Science, and their associated subfields, which form the basis for field recommendations.

An assessment scores dataset that contains test scores to assess the learner's proficiency levels and to aid recommendations based on their overall scores and related attributes.

A validation and benchmarking checklist to ensure alignment with academic (Malaysia Education Blueprint: MEB alignment considers digital literacy, ICT integration, computational thinking, AI concepts, learning outcomes, student-centered learning, career relevance, and related educational requirements) and industrial requirements (digital and workplace skills, problem-solving, emerging technologies, practical application, career pathways, skill progression, professional standards, and Industry 4.0 relevance). These checklists are dynamic rather than fixed and can be updated

based on the target educational standard, industry demand, computing field, or recommendation goal.

B. Learner Profiling Process

The system captures students' demographic, behavioral, and other personalized learning attributes, and the ML label encoder transforms categorical data, such as gender and learning style, into numeric codes for use by all ML models. The encoded profiles then form feature vectors, which the ML models use. Fig. 2 illustrates the process, as shown in Eq. (1)

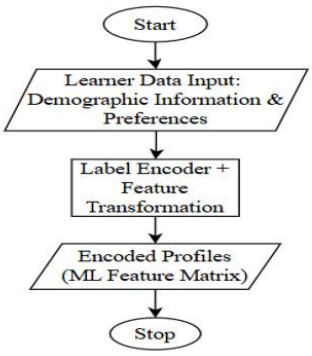


Fig. 2. Learner profiling process.

$$X_i = [x_{i1}, x_{i2}, \ldots x_{in}] \in \mathbb{R}^n \quad (1)$$

where, X_i represents the ith learner across n attributes, $\mathbb{R}^n$ is the n-dimensional real space.

The process flows as follows: learner demographic and other personalized learning input data → encoding → ML feature matrix. The profiling stage transforms qualitative descriptors into quantitative representations. For instance, "Male/Female" is encoded as a binary variable (0/1), and settlement data such as "Urban/Rural" is similarly represented as (0/1). This preprocessing ensures compatibility with ML algorithms, which operate exclusively on numerical data spaces. The resulting feature vector X_i represents the ith learner across n attributes and serves as input to the assessment and recommendation layers.

C. Assessment and Skill Level Evaluation

At this stage, each student takes predefined quizzes covering all the basic concepts of the identified Computing subfields and careers. The scores quantify proficiency and serve as learning features. Skill-level labels (beginner, intermediate, and advanced) are derived from performance thresholds. Fig. 3 shows the representation of the process, and Eq. (2) and (3).

$$Score_{ij} = \frac{CorrectAnswers}{TotalQuestions} \times 100 \quad (2)$$

where, $Score_{ij}$ represents the assessment score of the ith learner in the jth assessment, calculated as the percentage of correctly answered questions.

$$S_i = [S_{i1}, S_{i2}, \ldots, S_{im}] \quad (3)$$

where, S_i denotes the assessment score vector of the ith learner, consisting of m assessment scores.

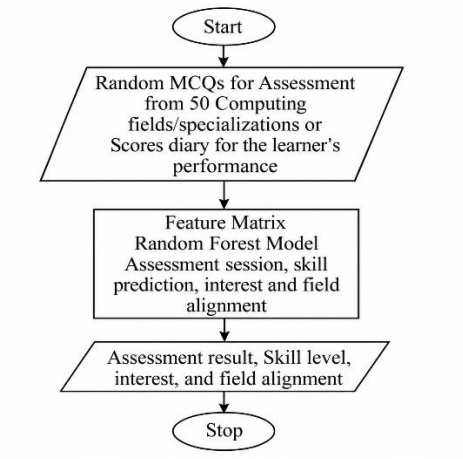


Fig. 3. Learners' assessment process.

The pipeline goes in this format:

Pipeline: Questions → Scores → Feature Matrix → Skill Prediction.

The skill evaluation process translates assessments into machine-readable performance metrics. Each dimension S_{ij} corresponds to the learner's score in one subfield. The dataset thus becomes a labeled multi-dimensional vector space where Random Forest classifiers can infer latent skill categories (Beginner, Intermediate, Advanced), and a supervised learning model is also incorporated.

D. Field Prediction Process

This stage predicts the optimal computing subfield for each student, models non-linear relationships between scores and fields or specializations, and handles multi-class classification through aggregated decision voting. Fig. 4 shows the process flow.

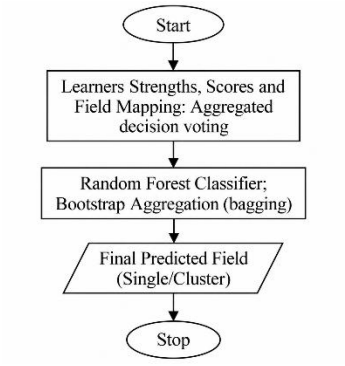


Fig. 4. Field prediction process.

This stage identifies the Computing field most aligned with a student's strengths. Each decision tree produces a class label such as Data Science, Cybersecurity, or Networking. The model aggregates these predictions through majority voting. Random Forest's bootstrap aggregation (bagging) strategy reduces variance, making predictions more stable across heterogeneous learners. The prediction process can be represented mathematically by Eq. (4)

$$y_{(pred)} = \arg\max_{y} \sum_{t=1}^{T} I\left(h_t(X) = y\right) \quad (4)$$

where, y_{pred} is the prediction label, y is the class, t is the individual tree, and **T** is the total number of trees; I is the indicator function, h is the hypothesis function, and **X** is the input vector.

E. Learner Proficiency Level and Framework Performance

This stage predicts the learner's overall proficiency level. The Standard-Scaler ensures uniform feature scaling. Thus, the framework performance is evaluated using accuracy, precision, recall, and F1-score, as shown in Eq. (5), (6), (7), and (8).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

where, Precision is the proportion of correct positive predictions, Recall is the proportion of actual positives that were correctly identified, F1 is the harmonic mean of Precision and Recall, TP denotes true positives, FP denotes false positives, TN denotes true negatives, and FN denotes false negatives. The process is presented as shown in Fig. 5.

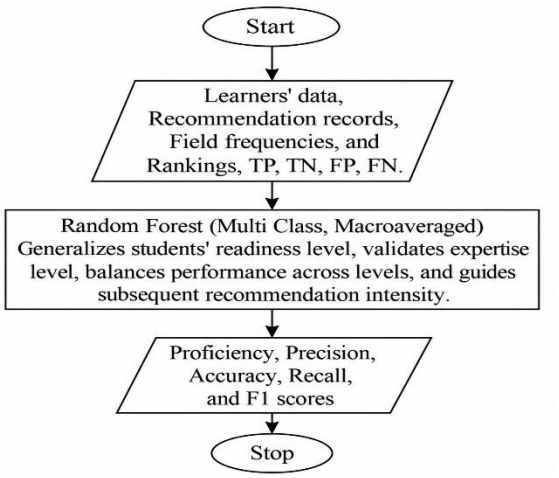


Fig. 5. Learner proficiency and framework performance.

Skill classification generalizes the student's readiness level. The macro-averaged F1 ensures balanced performance across all levels, even when classes are imbalanced. This stage ascertains how well the assessment vectors predict true expertise levels, guiding subsequent recommendation intensity for basic versus advanced materials.

F. Field Recommendations

The framework recommends fields and learning materials based on predicted outcomes, utilizing data-driven predictions with curated educational resource databases. The recommendation combines ML-based inference with dynamic, rule-based content retrieval. The justification for choosing Random Forest is its interpretability, robustness with small datasets (N $\leqslant$ 500), and resilience to feature correlations, which is important given the conceptual overlap between fields. Unlike deep learning alternatives, this approach requires no GPU infrastructure and yields clear feature importance rankings, supporting explainability in advising contexts [67]. Rule-based filtering is also utilized, as in Eq. (9) and Fig. 6.

$$Query = urllib.quote\_plus(Field\_Name) \quad (9)$$

where, $Query$ is the URL-safe search string, $urllib.quote\_plus$ is a Python function that encodes spaces as '+' and special characters for use in web URLs, and Field Name is the full, human-readable name of the predicted computing field, such as Software Engineering, Data Science, and Cybersecurity.

In summary, this component converts framework predictions into educational pathways. Once the system identifies a field, for instance, "Cloud Computing," it queries multiple online repositories such as YouTube, Coursera, EdX, and Google Scholar to dynamically generate URL queries linking to relevant, high-quality materials that support the user's learning journey.

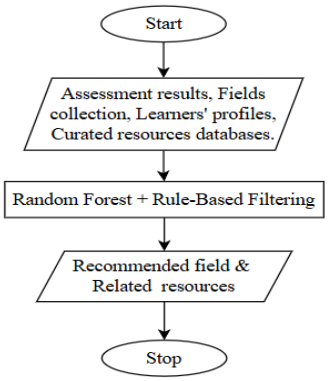


Fig. 6. Field recommendation process.

G. Resource Validation and Benchmarking

This stage is vital to ensure that the recommended materials align with both academic standards (the Malaysia Education Blueprint or MEB 2013-2025) and industry requirements specific to each computing-related career or field. Recommendations will be drawn from the top links on each specified platform, checking for keywords that match the criteria provided. The recommended material is considered suitable if it meets at least six criteria in the list. This process serves as a quality-control layer that ensures the recommended materials are relevant. This process is rule-based and deterministic, not reliant on a predictive ML algorithm, and aims to enhance the framework's performance. The mathematical process is expressed as in Eq. (10), as shown in Fig. 7.

$$Validated = \left(MEB_{yes} + Industry_{\text{yes}}\right) >= 6 \quad (10)$$

NB. To validate each recommended resource effectively, we need access to defined learning sources such as Khan

Academy, Coursera, edX, MIT, etc. So this work relies on free, publicly available web resources as a proof of concept. Table I presents examples of dynamically generated resource-search links based on the recommended computing field. These links support resource discovery and should not be interpreted as individually validated learning resources.

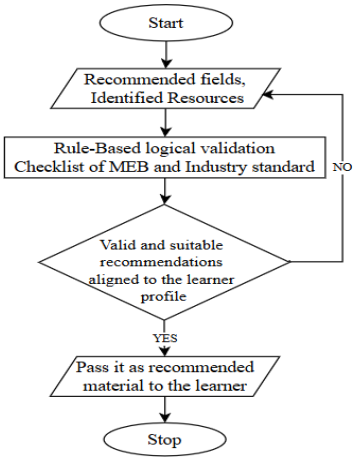


Fig. 7. Benchmarking and validation process flow.

H. Resource Generation and Retrieval

The system generates search queries using sanitized field names, then retrieves categorized resources such as videos, websites, and PDFs from multiple web sources. This action enhances the user experience by dynamically linking structured knowledge sources. Involves NLP-based string cleaning and automated URL generation for search queries. The flow is as in Fig. 8.

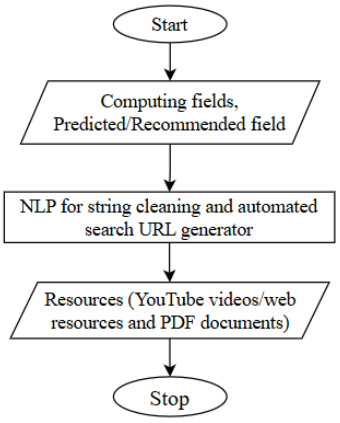


Fig. 8. Resource generation and retrieval.

NLP is used for dynamic query generation, including cleaning and encoding field names (e.g., removing parentheses and symbols), to ensure semantic accuracy in web search results. The objective is to maximize content relevance for each predicted field, as mathematically represented in Eq. (11).

$$URL = Base + urllib.parse.quote(Field) \quad (11)$$

where, URL denotes the generated link, Base represents the platform's search endpoint, $urllib.parse.quote$ is a function that URL-encodes the field name, and Field refers to the full subfield name (e.g., database engineering, multimedia technology, etc.).

I. Feedback and Logging

Collects user ratings and engagement metrics for post-recommendation. It also gathers log interaction data to retrain ML models for future recommendations. It adopts reinforcement learning principles for adaptive performance improvement and utilizes a feedback updater to achieve this, represented mathematically in Eq. (12).

$$Loss_{t+1} = \eta \frac{\partial Loss}{\partial \theta} \quad (12)$$

where, $Loss_{t+1}$ is the updated loss after iteration. $t+1$, η is the learning rate, and $\frac{\partial Loss}{\partial \theta}$ represents the gradient of the loss with respect to the model parameters. The flow proceeds as follows: User Feedback → Model Update → New Recommendations. This looping flow is essential for continuous improvement. It treats user feedback as a reward signal, which optimizes the model's parameters through gradient-based updates. Conceptually, this forms a self-correcting intelligent tutor that adapts and evolves based on learner interactions. Fig. 9 illustrates the process.

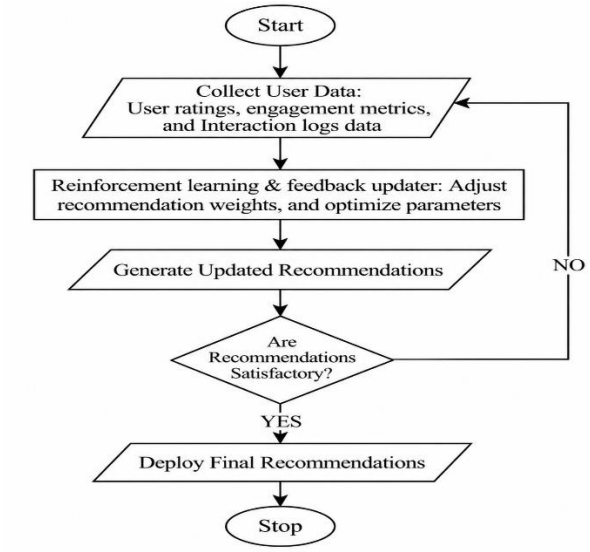


Fig. 9. Feedback loop and logging.

J. Overall Evaluation Accuracy of ComReF

At this stage, the framework is evaluated using the previously computed F1-score, Precision, and recall to determine overall accuracy and ensure balanced predictions across all identified computing fields. The evaluation process is conducted using $sklearn.metrics.classificationreport$ with the overall accuracy represented in Eq. (13) and illustrated in Fig. 10, respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

where, Accuracy denotes the proportion of correct predictions, TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively, measuring the overall correctness across all classes. This final

stage summarizes system performance. The F1-score provides a harmonic balance between Precision and Recall, which is critical for recommendation tasks. A high macro-average F1-score suggests that the framework is suitable for large-scale deployment, as it reflects consistent performance across all categories. Fig. 10 depicts the overall evaluation process.

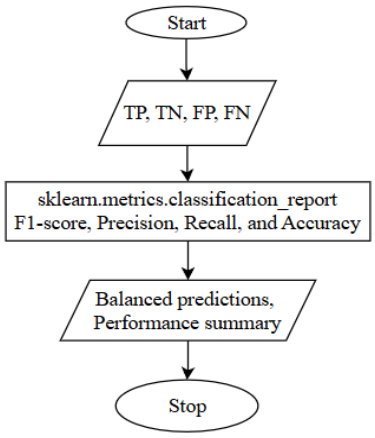


Fig. 10. Overall evaluation process.

IV. Implementation and Results

This section explains the implementation of the proposed ComReF, starting with the learner profiling process, followed by learning-resource recommendations, a report on classification performance, prototyping, and a comparison of the chosen Random Forest with three classification models. The dataset comprised 500 instances across 50 computing-field classes. The process used an 80:20 train-test split, resulting in 400 training and 100 testing instances; the same split was used across all evaluated classifiers to ensure a consistent comparison. No cross-validation was applied. Some computing fields/subfields were sparsely represented, with only 1 to 3 instances in the test set; therefore, interpreting the corresponding class-level performance metrics cautiously.

A. Learner Profiling

The framework's recommendations center on the learner profile, which drives the process. Thus, the profile includes demographic data and learner preferences that ChatGPT generates from prompts tailored to the expected diversity of learners in the Malaysian context.

	S/No	Name	Gender	Grade	Race	Settlement	Skill Level
0	1.0	Lim G.	Female	12.0	Chinese	Urban	Advanced
1	2.0	Abdullah G.	Female	9.0	Malay	Rural	Beginner
2	3.0	Martin I.	Male	10.0	International	Urban	Beginner
3	4.0	Chan P.	Female	12.0	Chinese	Rural	Beginner
4	5.0	Naidu A.	Female	12.0	Indian	Urban	Intermediate

	Learning Attitude	Learning Motivation	Learning Goal	Learning Interest
0	Collaborative	Intrinsic	Innovation	Mobile Apps
1	Collaborative	Extrinsic	Innovation	Quantum Computing
2	Independent	Intrinsic	Innovation	AR/VR
3	Independent	Extrinsic	Skills	Cloud Computing
4	Independent	Intrinsic	Academic	AI & Machine Learning

	Learning Preference	Career Goals
0	Solitary	Computer Scientist, System Analyst, Researcher
1	Visual	IT Specialist, Systems Analyst
2	Kinesthetic	Computer Engineer, Embedded Systems Engineer
3	Reading/Writing	IS Analyst, Systems Manager
4	Kinesthetic	MIS Specialist, IT Manager

Fig. 11. Basic learner profile information.

Fig. 11 illustrates the profiling information, which includes the students' IDs, names, races, residences, perceived skill levels, learning attitudes, motivations, goals, interests, preferences, and career aspirations of the first five learners: Lim, Abdullah, Martin, Chan, and Naidu, and Fig. 12 then shows their respective recommendation report summaries: Lim for Cybersecurity, Abdullah for Data Science, Martin for Computer Engineering, Chan for Software Engineering, and Naidu for Computer Science. The full profile and recommendations are saved as a CSV file in a folder.

✅ Final Recommendation Report saved as Final_Recommendation_Report.csv

	student_id	predicted_field	predicted_skill_level
0	1	Cybersecurity	Intermediate
1	2	Data Science	Intermediate
2	3	Computer Engineering	Beginner
3	4	Software Engineering	Intermediate
4	5	Computer Science	Beginner

	career_goals	resources_validated	validation_yes_total
0	Cybersecurity Analyst	True	16
1	Data Scientist	True	16
2	Embedded Systems Engineer	True	16
3	Software Engineer	True	16
4	Computer Scientist	True	16

Fig. 12. Recommendation summary related to the student's profile.

B. Recommended Learning Resources

The recommended learning resources include YouTube video links, PDFs, and educational websites, all aligned with the learner's profile and assessment results in the suggested learning field. Based on the profiles in Fig. 11 and 12 and using Abdullah G. as an example for whom Data Science was recommended, Table II shows his recommended learning resources, including YouTube links, websites, and PDFs. The table was formatted for clarity, as the original display in the Anaconda environment looks complex/long, and links may not look well-formatted here.

TABLE II. Recommended Resources for Abdullahi G., Data Science

1. Abdullah G. ---------- Data Science		
S/No.	**Resource Type**	**Link**
1	YouTube	https://www.youtube.com/results?search_query=Data+Science+tutorial
2	YouTube	https://www.youtube.com/results?search_query=Data+Science+course
3	YouTube	https://www.youtube.com/results?search_query=Data+Science+lecture
4	YouTube	https://www.youtube.com/results?search_query=Data+Science+advanced
5	Website	https://en.wikipedia.org/wiki/Data+Science
6	Website	https://www.coursera.org/search?query=Data+Science
7	Website	https://www.edx.org/search?q=Data+Science
8	Website	https://www.kaggle.com/search?q=Data+Science
9	Website	https://www.researchgate.net/search/publication?q=Data+Science
10	Website	https://www.ibm.com/search?lang=en&cc=us&q=Data+Science
11	PDF	https://scholar.google.com/scholar?q=Data+Science+filetype:pdf
12	PDF	https://www.google.com/search?q=filetype:pdf+Data+Science+course
13	PDF	https://www.google.com/search?q=filetype:pdf+Data+Science+lecture+notes
14	PDF	https://www.google.com/search?q=filetype:pdf+Data+Science+handbook

C. Classification Performance

This section presents the classification performance report of ComReF. The predictive performance was evaluated using standard metrics, including Precision, Recall, and F1-score, computed per class and aggregated using macro and weighted averages. For the best field prediction, ComReF achieved an overall accuracy of 64% with a weighted F1-score of 0.59, indicating moderate performance across the subfields/careers in Computing. Notably, several classes (1, 2, 3, 9, 13, 16, 18, . . . , 34, 36, and 49) achieved perfect or near-perfect F1-scores (F1 ≥ 0.80), demonstrating strong generalization for well-represented fields. However, classes (4, 6, 7, 12, 17, 19, 21, 22, . . . , 40, 42, and 43) recorded comparatively lower performance metrics due to limited sample representation (support = 1–3). This outcome aligns with the known behavior of multi-class classifiers under imbalanced class distributions and does not undermine the overall performance trends. In contrast, the skill-level prediction model achieved 100% accuracy (F1 score = 1); however, this reflects the uniformity of the test set, where all students exhibited similar/the same background skill levels. These results indicate that ComReF performs well when sufficient training data is available, highlighting the importance of data adequacy for future data collection and taxonomy refinement. Fig. 13 illustrates the classification results.

Best Field Prediction Report (Classes 0 – 30)

Class	Precision	Recall	F1-score	Support
0	0.50	1.00	0.67	2
1	1.00	1.00	1.00	2
2	0.75	1.00	0.86	3
3	1.00	1.00	1.00	1
4	0.00	0.00	0.00	1
5	0.67	0.67	0.67	3
6	0.00	0.00	0.00	2
7	0.00	0.00	0.00	3
8	0.25	0.50	0.33	2
9	1.00	1.00	1.00	2
10	0.33	1.00	0.50	1
11	0.50	0.33	0.40	3
12	0.00	0.00	0.00	2
13	1.00	1.00	1.00	2
14	0.50	0.50	0.50	2
15	0.67	1.00	0.80	4
16	1.00	1.00	1.00	2
17	0.00	0.00	0.00	2
18	0.00	1.00	0.00	4
19	0.00	0.00	0.00	1
20	1.00	0.50	0.67	2
21	0.00	0.00	0.00	1
22	0.00	0.00	0.00	1
23	0.00	0.00	0.00	1
24	0.00	0.00	0.00	1
25	0.00	1.00	0.00	1
26	0.50	1.00	0.67	1
27	0.67	1.00	0.80	2
28	1.00	1.00	1.00	3
29	0.67	0.67	0.67	3
30	0.50	0.33	0.40	3

Best Field Prediction Report (Classes 31 – 49)

Class	Precision	Recall	F1-score	Support
31	0.75	1.00	0.86	3
32	1.00	1.00	1.00	1
33	0.00	0.00	0.00	1
34	1.00	1.00	1.00	2
35	1.00	0.50	0.67	2
36	0.75	1.00	0.86	3
37	0.50	1.00	0.67	3
38	0.50	0.67	0.57	3
39	0.50	1.00	0.67	2
40	0.00	0.00	0.00	1
41	0.50	0.50	0.50	2
42	0.00	0.00	0.00	2
43	0.00	0.00	0.00	1
44	0.67	0.67	0.67	3
45	0.50	0.50	0.50	2
46	1.00	0.50	0.67	2
47	1.00	0.50	0.67	2
48	1.00	0.50	0.67	2
49	1.00	1.00	1.00	1
Accuracy			**0.64**	**100**
Macro avg	**0.52**	**0.58**	**0.53**	**100**
Weighted avg	**0.58**	**0.64**	**0.59**	**100**

Skill Level Prediction Report

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	100
Accuracy				**100**
Macro avg	1.00	1.00	1.00	**100**
Weighted avg	1.00	1.00	1.00	**100**

Fig. 13. The classification report.

Further implies that ComReF can accurately map student profiles to predicted or recommended fields and generate valid, skill-tagged learning resources across many platforms. Hence, the observed performance limitations, especially the low F1-scores for certain fields, come from data rather than design flaws. While the architecture is solid, rare or fragmented fields faced limited labeled representations. Possible enhancements of ComReF include merging similar niches, increasing data collection for underrepresented pathways, and implementing the pilot ComReF in live academic advising workflows to collect real-world labels and refine its calibration through iterative improvements.

D. Prototyping

This section elaborates on how the ComReF prototype is designed for recommendations in academic advising environments. It requires minimal infrastructure and offers broad accessibility. It also provides two main inference interfaces: batch processing through CSV uploads and real-time prediction through webform integration.

CSV Upload (Batch Mode): Users can upload a CSV file containing one or more student profiles. Each row must include a student ID and scores (1–5) for all computing fields, for instance, Software Engineering, Data Science, Cybersecurity, and Network Engineering. The system processes the file within seconds, with a mean inference time of 0.38 seconds per student. It then generates a structured CSV output that includes the predicted field and skill level, career goals, and resource links from platforms such as YouTube, Wikipedia, Coursera, edX, Kaggle, ResearchGate, and Google Scholar (PDFs).

Web Form (Real-Time Mode): For integration purposes, ComReF provides a lightweight API endpoint. Both interfaces are implemented entirely in Python using pandas and scikit-learn, ensuring full cross-platform compatibility (Windows, macOS, Linux) and eliminating any dependence on large language models (LLMs) or cloud-based inference. Table III shows a representative input-output pair for a student with strong aptitude in Cryptography, Cybersecurity, Software Engineering, and AI. The prototype then generates the output in Table IV.

TABLE III. SAMPLE INPUT FOR INFERENCE LEARNER

Input Feature	Assessment Scores
Cryptography	8
Cybersecurity	6
Software Engineering	5
Artificial Intelligence	4
…List	…

TABLE IV. THE INFERENCE OUTPUT

Output Component	Value
Predicted Field	Cryptography
Skill Level	Advanced
Career Goal	System security expert
Resources	Four YouTube Videos, Six Learning sites, and four PDF documents

E. Baseline Comparison: Model Selection Justification

To justify selecting the Random Forest algorithm, we benchmarked three widely used classifiers: Decision Tree (DT), Logistic Regression (LR), and Naïve Bayes (NB). We evaluated each model under the same conditions: the same feature preprocessing pipeline and scaling techniques, with standard scaling applied to Random Forest, Decision Tree, and Logistic Regression, and Min–Max scaling applied to Naïve Bayes. All models were trained and tested using the same data splits and evaluated with consistent performance metrics. As demonstrated in Table IV, Random Forest consistently outperforms all baseline models in field-level prediction, achieving an accuracy of 0.640 and a weighted F1-score of 0.586. In comparison, Decision Tree (weighted F1 = 0.174), Naïve Bayes (weighted F1 = 0.321), and Logistic Regression

(weighted F1 = 0.086) exhibit significantly lower performance. This is especially evident given the task's high dimensionality and multi-class nature, which challenges Logistic Regression. Furthermore, as the macro F1-score indicates, the selected approach is robust to class imbalance (see Table V).

TABLE V. CLASSIFICATION PERFORMANCE ON THE FIELD PREDICTION FOR MODEL JUSTIFICATION

Model	Accuracy	Weighted F1	Macro F1
Random Forest	0.64	0.586	0.529
Decision Tree	0.20	0.174	0.118
Logistic Regression	0.09	0.086	0.074
Naïve Bayes	0.37	0.321	0.252

From Table V, Random Forest achieved the highest overall performance; the difference between its weighted F1 (0.586) and macro F1 (0.529) indicates uneven performance across computing fields, particularly those with fewer samples. This suggests that class imbalance may have affected minority-field predictions. Notwithstanding, Random Forest achieved the highest macro F1 among all evaluated models, indicating comparatively better performance across classes. The moderate performance should be considered in the context of the 50-class prediction setting and limited observations per class; unequal class representation may also contribute to the lower macro F1-score relative to the weighted F1-score. Future evaluation will therefore consider more balanced field representation to improve performance for underrepresented fields. Under controlled experimental conditions, Random Forest consistently outperformed the benchmark models in accuracy, weighted F1, and macro F1, supporting its selection as the primary classifier for ComReF.

V. DISCUSSION

ComReF is a learner-centered AI application for computing education that provides computing-field recommendations and related learning resources for secondary and pre-university learners. It is particularly intended to support learners who require early guidance on computing specializations before progressing to university-level studies. The prototype demonstrates the feasibility of combining learner profiles and assessment performance for personalized computing-field recommendations while remaining lightweight and interpretable. The following subsections discuss the findings in relation to previous research, feedback, and practical implications, and ComReF's strengths, limitations, and ethical considerations.

A. Findings, Feedback and Practical Implications

The findings agree with previous studies on AI-supported educational recommendations. Algarni and Sheldon [2] demonstrated the relevance of recommender systems for course selection, while Narimani and Barberà [37] emphasized learner profiling in personalized recommendations. Similarly, Wang *et al.* [43] showed the value of machine learning for personalized feedback. ComReF extends these approaches by integrating learner profiling, computing-field prediction, skill-level identification, and learning resources to support computing specialization at secondary and pre-university levels. ComReF also incorporates a feedback loop in which learner performance and feedback can support reassessment and refinement of field and resource recommendations. In practice, it is intended to complement teacher and academic-adviser guidance, with periodic reassessment as learners' interests, skills, and performance change.

B. Strengths of the Proposed Work

Compared with other recommendation works, ComReF's major strength is its end-to-end integration of diagnostic assessment, career-path prediction, and learning-resource generation. ComReF's recommendations offer a complete learning itinerary that can reduce the major gap noted in prior studies, particularly the persistent disconnect between prediction and pedagogy. ComReF ensures that all recommendations are fully accessible to learners. In addition, ComReF aligns with the demand for explainable and trustworthy AI in education, strengthening confidence among students, advisors, and institutional stakeholders. ComReF is lightweight and suitable for computing educational platforms that require an advising module. In terms of adaptation, ComReF offers three major advantages due to its lightweight architecture:

- No GPU or cloud dependency: It runs efficiently on a standard CPU using less than 500 MB of RAM; thus, it is suitable even for local laptops and low-resource environments.
- Auditable: All processes rely on interpretable ML and rule-based logic; hence, it avoids the opacity of black-box LLMs.
- Preserve privacy: Student data is processed within the local environment without external API calls apart from accessing the generated resource URLs.

C. Limitations

However, ComReF is data-dependent as it relies on a labeled dataset containing students' profiles (demographic information and learning preferences), which is not always readily available. Thus, a primary challenge lies in acquiring sufficient labeled data. Future work could address this limitation by adopting semi-supervised or active learning strategies to boost the labeling process. Another limitation concerns the use of static resource templates, as the URL structures are hardcoded for major platforms. If a platform modifies its query syntax, for example, Coursera changing from: $?query = to?q =$ ", then the generated links may become invalid. An alternative approach would involve dynamic API integration or automated monitoring to increase dynamism in such cases.

D. Ethical Considerations

The first ethical issue involves the training data common in all supervised systems. Thus, ComReF inherits any biases present in its training data. If historical data has favored certain fields, such as Web Development over Bioinformatics, ComReF may spread these imbalances. Stratified sampling is necessary to mitigate this issue and evaluate performance across fields. However, fairness audits remain crucial to ensure balanced recommendations for learners' learning journey.

Another issue is the absence of equity across fields. Emerging fields such as Neurotechnology and Climate Informatics may have fewer learning resources available on mainstream platforms, potentially disadvantaging students who receive recommendations for these areas. To resolve this issue, ComReF provides a checklist protocol that flags recommendations with fewer than three available resources, indicating low-resource fields. Student agency versus algorithmic determinism is a concern. ComReF offers guidance; recommendations must not be perceived as prescriptive. In other words, all outputs clearly state that recommendations are suggestions, not mandates for learners. Future versions could consider counterfactual explanations like: "If you improved your score in Generative AI by 1 point, you would be recommended AI Ethics," to encourage exploration and give students more autonomy in their choices. Hence, ComReF demonstrates that lightweight, rule-based AI can deliver high-impact educational value when designed with pedagogical intent, in this case, for secondary or pre-university students in the Computing field. It offers a feasible alternative to learning via AI-driven recommendations, rather than the existing rigid approach.

VI. Conclusion and Future Work

In conclusion, ComReF provides a personalized approach for guiding pre-university learners toward suitable computing fields using their profiles and assessment performance. It combines learner profiling, feature encoding, machine-learning prediction, skill-level estimation, and learning-resource recommendation. Among the evaluated classifiers, Random Forest performed best, with an accuracy of 0.64, weighted F1 of 0.586, and macro F1 of 0.529. The lower macro F1 reflects performance differences across computing fields, particularly those with limited representation. Skill-level prediction recorded an F1 of 1.00; however, only one skill-level class appeared in the test set, so this result does not represent discrimination across different skill levels.

ComReF contributes by integrating computing-field prediction, learner characteristics, skill level, and relevant learning resources into a single recommendation process for early computing specialization. Future work will extend recommendations to subfields within broad areas such as Software Engineering, AI, and Data Science and explore multi-label prediction for ranked field choices. Further evaluation will involve real learner profiles and integration with academic advising platforms to examine learner satisfaction, retention, career alignment, and industry relevance. LLMs may also be considered for resource summarization without replacing the framework's core prediction mechanism, while releasing ComReF as an open-source Python package could support reproducibility, improvement, and wider adoption.

Acknowledgment

This article is a deliverable of the PhD thesis of the first author's PhD thesis, completed under the supervision of the co-authors. Artificial intelligence tools were also used in language improvement, organization, sample demographic data generation, and refinements.

References

[1] E. Ugwu and R. O. M. Ugwoke, "An Assessment of Choices of Areas of Specialization of Computer Education Students in Fields of Information and Communication Technology in Universities in South Eastern Nigeria," Sapientia Foundation Journal of Education, Sciences and Gender Studies (SFJESGS) vol. 2, no. 1, pp. 80–89, 2020.

[2] S. Algarni and F. Sheldon, "Systematic Review of Recommendation Systems for Course Selection," Mach. Learn. Knowl. Extr., vol. 5, no. 2, pp. 560–596, 2023, doi: 10.3390/make5020033.

[3] L. Huang, C. D. Wang, H. Y. Chao, J. H. Lai, and P. S. Yu, "A Score Prediction Approach for Optional Course Recommendation via Cross-User-Domain Collaborative Filtering," IEEE Access, vol. 7, pp. 19550–19563, 2019, doi: 10.1109/ACCESS.2019.2897979.

[4] H. Abu-Rasheed, C. Weber, and M. Fathi, "Context-based learning: a survey of contextual indicators for personalized and adaptive learning recommendations – a pedagogical and technical perspective," Front. Educ., vol. 8, no. July, 2023, doi: 10.3389/feduc.2023.1210968.

[5] S. Cha, M. Loeser, and K. Seo, "The Impact of AI-Based Course-Recommender System on Students' Course-Selection Decision-Making Process," Appl. Sci., vol. 14, no. 9, pp. 1–14, 2024, doi: 10.3390/app14093672.

[6] L. Labadze, M. Grigolia, and L. Machaidze, "Role of AI chatbots in education: systematic literature review," Int. J. Educ. Technol. High. Educ., vol. 20, no. 1, pp. 1–17, 2023, doi: 10.1186/s41239-023-00426-1.

[7] S. Ray and A. Sharma, "A collaborative filtering based approach for recommending elective courses," Communications in Computer and Information Science, vol. 141 CCIS. pp. 330–339, 2011. doi: 10.1007/978-3-642-19423-8_34.

[8] C. Vandelanotte et al., "Increasing physical activity using a just-in-time adaptive digital assistant supported by machine learning: a novel approach for hyper-personalised mHealth interventions," J. Biomed. Inform., vol. 144, no. January, p. 104435, 2023, doi: 10.1016/j.jbi.2023.104435.

[9] S. Tambuskar, "Challenges and Benefits of 7 ways Artificial Intelligence in Education Sector," Rev. Artif. Intell. Educ., vol. 3, no. 00, p. e03, 2022, doi: 10.37497/rev.artif.intell.education.v3i00.3.

[10] V. E. Kulugh, K. P. Aondover, and A. S. Faki, "Artificial Intelligence-Powered Personalised Career Guidance System," Dutse J. Pure Appl. Sci., vol. 11, no. 1c, pp. 92–104, 2025, doi: 10.4314/dujopas.v11i1c.11.

[11] G. J. Hwang, H. Y. Sung, S. C. Chang, and X. C. Huang, "A fuzzy expert system-based adaptive learning approach to improving students' learning performances by considering affective and cognitive factors," Comput. Educ. Artif. Intell., vol. 1, no. August, p. 100003, 2020, doi: 10.1016/j.caeai.2020.100003.

[12] S. Bisaso, W. Wasswa, and G. Muhumuza, "Towards the Application of an NLP-driven Conversational Model for Efficient and Affective Career Guidance for Students : A Scoping Review," vol. 3, no. 3, pp. 240–253, 2025, doi: 10.59324/ejaset.2025.3(3).16.

[13] F. Redmond, With a Rise in Computing Disciplines Comes a Greater Choice of Computing Degrees in Higher Education, vol. 1, no. 1. Association for Computing Machinery, 2022. doi: 10.1145/3564721.3565946.

[14] R. W. Lent, A. M. Lopez, F. G. Lopez, and H. Sheu, "Social cognitive career theory and the prediction of interests and choice goals in the computing disciplines q," vol. 73, pp. 52–62, 2008, doi: 10.1016/j.jvb.2008.01.002.

[15] L. J. Sax, K. N. S. Newhouse, J. Goode, T. M. Nakajima, M. Skorodinsky, and M. Sendowski, "Can Computing Be Diversified on 'principles' Alone? Exploring the Role of AP Computer Science Courses in Students' Major and Career Intentions," ACM Trans. Comput. Educ., vol. 22, no. 2, 2022, doi: 10.1145/3479431.

[16] Abdul Razak, N. H., Ismail, S. N., Mahadzir, N. H., Idrus, Z., & Ramli, R. (2024). Understanding The Challenges of Teaching Computer Science Subjects from School Teachers' Perspective. Journal of Technology and Operations Management, 19(1), 1-10. https://doi.org/10.32890/jtom2024.19.1.1

[17] M. A. Nasir, R. Talib, A. A. Latif, M. F. A. Hanid, and H. A. Harmeni, "STEM Productive Learning of Lower Secondary School in Southern Zone, Malaysia," Int. J. Learn. Teach. Educ. Res., vol. 22, no. 2, pp. 281–301, 2023, doi: 10.26803/ijlter.22.2.16.

[18] 2018 National Academies of Sciences, Washington, DC, Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments (2018). 2018. doi: 10.17226/24926.

[19] P. M. Alexander et al., "Factors Affecting Career Choice : Comparison Between Students from Computer and Other Disciplines," pp. 300–315, 2011, doi: 10.1007/s10956-010-9254-3.

[20] H. Lu, "Research on Computer Aided Learning System Based On Reinforcement Learning Algorithm," Procedia Comput. Sci., vol. 243, pp. 472–481, 2024, doi: 10.1016/j.procs.2024.09.058.

[21] J. Mahadeo, Z. Hazari, and G. Potvin, "Developing a Computing Identity Framework : Understanding Computer Science and Information Technology Career Choice," vol. 20, no. 1, pp. 1–14, 2020.

[22] A. Bollin, S. Pasterk, M. Kesselbacher, E. Reci, M. Wieser, and N. Lobnig, "HCI in K12 computer science education - Using HCI as a topic and a didactic tool," ACM Int. Conf. Proceeding Ser., vol. 1, no. 1, pp. 1–14, 2021, doi: 10.1145/3464385.3464717.

[23] G. M. Dhananjaya, R. H. Goudar, A. A. Kulkarni, V. N. Rathod, and G. S. Hukkeri, "A Digital Recommendation System for Personalized Learning to Enhance Online Education: A Review," IEEE Access, vol. 12, no. March, pp. 34019–34041, 2024, doi: 10.1109/ACCESS.2024.3369901.

[24] J. Yao, "Exploring Experiential Learning: Enhancing Secondary School Chemistry Education Through Practical Engagement and Innovation," J. Educ. Humanit. Soc. Sci., vol. 22, pp. 475–484, 2023, doi: 10.54097/ehss.v22i.12508.

[25] M. J. Bishop, E. Boling, J. Elen, and V. Svihla, "Handbook of Research in Educational Communications and Technology: Learning Design: Fifth Edition," Handb. Res. Educ. Commun. Technol. Learn. Des. Fifth Ed., pp. 1–897, 2020, doi: 10.1007/978-3-030-36119-8.

[26] L. S. H. Cardenas, L. Castano, C. C. Guzman, and J. P. N. Alvarez, "Personalised learning model for academic leveling and improvement in higher education," Australas. J. Educ. Technol., vol. 38, no. 2, pp. 70–82, 2022, doi: 10.14742/ajet.7084.

[27] J. M. Wing, "Computational thinking's influence on research and education for all Influenza del pensiero computazionale nella ricerca e nell'educazione per tutti," Ital. J. Educ. Technol., vol. 25, no. 2, pp. 7–14, 2017, doi: 10.17471/2499-4324/922.

[28] S. Paek and N. Kim, "Analysis of worldwide research trends on the impact of artificial intelligence in education," Sustain., vol. 13, no. 14, 2021, doi: 10.3390/su13147941.

[29] Y. Hachaichi, A. E. Khedr, M. A. Belal, and A. S. Elbhrawy, "A Predictive Model for Personalized Course Advising Using Data Mining Techniques," IEEE Access, vol. 13, no. September, pp. 163618–163626, 2025, doi: 10.1109/ACCESS.2025.3600578.

[30] G. Bilquise, S. Ibrahim, and S. M. Salhieh, "Investigating student acceptance of an academic advising chatbot in higher education institutions," Educ. Inf. Technol., vol. 29, no. 5, pp. 6357–6382, 2024, doi: 10.1007/s10639-023-12076-x.

[31] M. Tavakoli, A. Faraji, J. Vrolijk, M. Molavi, S. T. Mol, and G. Kismihók, "An AI-based open recommender system for personalized labor market driven education," Adv. Eng. Informatics, vol. 52, no. November 2021, p. 101508, 2022, doi: 10.1016/j.aei.2021.101508.

[32] Y. Lin et al., "A Comprehensive Survey on Deep Learning Techniques in Educational Data Mining," Data Sci. Eng., 2025, doi: 10.1007/s41019-025-00303-z.

[33] W. Cui, Z. Xue, and K. P. Thai, "Performance Comparison of an AI-Based Adaptive Learning System in China," Proc. 2018 Chinese Autom. Congr. CAC 2018, pp. 3170–3175, 2018, doi: 10.1109/CAC.2018.8623327.

[34] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under Concept Drift: A Review," IEEE Trans. Knowl. Data Eng., vol. 31, no. 12, pp. 2346–2363, 2019, doi: 10.1109/TKDE.2018.2876857.

[35] F. Altares, "Development and Validation of a Supplemental Learning Resource in Chemistry in Conversational Filipino," J. Interdiscip. Perspect., vol. 2, no. 1, pp. 71–86, 2024, doi: 10.69569/jip.2024.0010.

[36] Y. Gong, X. Li, and L. e. Wang, "FedMBC: Personalized federated learning via mutually beneficial collaboration," Computer Communications, vol. 205. pp. 108–117, 2023. doi: 10.1016/j.comcom.2023.04.012.

[37] A. Narimani and E. Barberà, "Extracting Course Features and Learner Profiling for Course Recommendation Systems: A Comprehensive Literature Review," Int. Rev. Res. Open Distrib. Learn., vol. 25, no. 1, pp. 197–225, 2024, doi: 10.19173/irrodl.v25i1.7419.

[38] P. Duong Thanh Tran, K. Bao Vu, M. Nhat Doan, Q. Truc Dang, Q. Viet Dang, and T. Trung Ho, "Personalized learning paths recommendation system with collaborative filtering and content-based approaches," Sci. Technol. Dev. J. - Econ. - Law Manag., vol. 8, no. 2, pp. 5243–5253, 2024, doi: 10.32508/stdjelm.v8i2.1370.

[39] X. Wei and H. Jia, "A Review of the Application of Artificial Intelligence in the Virtual Learning Environment," Proc. - 2021 10th Int. Conf. Educ. Innov. through Technol. EITT 2021, pp. 79–82, 2021, doi: 10.1109/EITT53287.2021.00024.

[40] Y. Lian, "Smart education: Education reform in the age of intelligence," ACM Int. Conf. Proceeding Ser., no. 4, pp. 41–45, 2021, doi: 10.1145/3502434.3502478.

[41] N. M. Ariff, B. L. K. Bang, S. Nadarajan, and M. H. M. Nor, "Educational Recommender Systems: a Bibliometric Analysis for the Period 2002 – 2022," J. Qual. Meas. Anal., vol. 20, no. 2, pp. 197–215, 2024, doi: 10.17576/jqma.2002.2024.14.

[42] R. Shafique, W. Aljedaani, F. Rustam, E. Lee, A. Mehmood, and G. S. Choi, "Role of Artificial Intelligence in Online Education: A Systematic Mapping Study," IEEE Access, vol. 11, no. June, pp. 52570–52584, 2023, doi: 10.1109/ACCESS.2023.3278590.

[43] X. Wang, L. Zhang, and T. He, "Learning Performance Prediction-Based Personalized Feedback in Online Learning via Machine Learning," Sustain., vol. 14, no. 13, 2022, doi: 10.3390/su14137654.

[44] T. Doleck, D. J. Lemay, R. B. Basnet, and P. Bazelais, "Predictive analytics in education: a comparison of deep learning frameworks," Educ. Inf. Technol., vol. 25, no. 3, pp. 1951–1963, 2020, doi: 10.1007/s10639-019-10068-4.

[45] L. Zhang, J. D. Basham, and R. A. Carter, "Measuring personalized learning through the Lens of UDL: Development and content validation of a student self-report instrument," Stud. Educ. Eval., vol. 72, no. September 2020, p. 101121, 2022, doi: 10.1016/j.stueduc.2021.101121.

[46] E. Frankford, C. Sauerwein, P. Bassner, S. Krusche, and R. Breu, "AI-Tutoring in Software Engineering Education Experiences with Large Language Models in Programming Assessments," Proc. - Int. Conf. Softw. Eng., pp. 309–319, 2024, doi: 10.1145/3639474.3640061.

[47] X. Chen et al., "Set-to-Sequence Ranking-Based Concept-Aware Learning Path Recommendation," Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023, vol. 37, pp. 5027–5035, 2023, doi: 10.1609/aaai.v37i4.25630.

[48] Z. Xiong et al., "A review of data mining in personalized education: Current trends and future prospects," … Digit. Educ., 2024, [Online]. Available: https://link.springer.com/article/10.3868/s110-009-024-0004-9%0Ahttps://arxiv.org/pdf/2402.17236

[49] Y. Cen, "EGRec : A MOOCs Course Recommendation Model Based on Knowledge Graphs EGRec : A MOOCs Course Recommendation Model Based on Knowledge Graphs," pp. 0–24, 2025.

[50] J. Jeevamol and V. G. Renumol, "An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem," Educ. Inf. Technol., vol. 26, no. 4, pp. 4993–5022, 2021, doi: 10.1007/s10639-021-10508-0.

[51] S. Benhamdi, A. Babouri, and R. Chiky, "Personalized recommender system for e-Learning environment," Educ. Inf. Technol., vol. 22, no. 4, pp. 1455–1477, 2017, doi: 10.1007/s10639-016-9504-y.

[52] D. Lee, Y. Huh, C. Y. Lin, and C. M. Reigeluth, "Personalized Learning Practice in U.S. Learner-Centered Schools," Contemp. Educ. Technol., vol. 14, no. 4, 2022, doi: 10.30935/cedtech/12330.

[53] I. Khan, A. R. Ahmad, N. Jabeur, and M. N. Mahdi, "An artificial intelligence approach to monitor student performance and devise preventive measures," Smart Learn. Environ., vol. 8, no. 1, 2021, doi: 10.1186/s40561-021-00161-y.

[54] J. Kim, A. Leftwich, and D. Castner, Beyond teaching computational thinking: Exploring kindergarten teachers' computational thinking and computer science curriculum design considerations, vol. 29, no. 12. Springer US, 2024. doi: 10.1007/s10639-023-12406-z.

[55] K. Aguar, H. R. Arabnia, J. B. Gutierrez, W. D. Potter, and T. R. Taha, "Making CS inclusive: An overview of efforts to expand and diversify CS education," Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016, pp. 321–326, 2017, doi: 10.1109/CSCI.2016.0067.

[56] S. W. Gundersen and G. Lampropoulos, "Using Serious Games and Digital Games to Improve Students' Computational Thinking and Programming Skills in K-12 Education: A Systematic Literature Review," Technologies, vol. 13, no. 3, 2025, doi: 10.3390/technologies13030113.

[57] S. Abdelhamid, J. Bangura, and S. Shah, "Advisely: AI-Powered Academic Advising Using Large Language Models (LLMs)," in New Perspectives in Science Education - International Conference, 2025.

[58] M. Van Mechelen et al., "Emerging Technologies in K–12 Education: A Future HCI Research Agenda," ACM Trans. Comput. Interact., vol. 30, no. 3, pp. 1–40, 2023, doi: 10.1145/3569897.

[59] J. Liu, M. Mitsui, N. J. Belkin, and C. Shah, "Task , Information Seeking Intentions , and User Behavior : Toward A Multi-level Understanding of Web Search," pp. 123–132, 2019, doi: 10.1145/3295750.3298922.

[60] M. R. Da Silva, L. L. Leite, and G. L. Garcia Costa Dos Santos, "Algorithms for the Development of Adaptive Web Interfaces: A Systematic Literature Review," 2022 17th Lat. Am. Conf. Learn. Technol. LACLO 2022, 2022, doi: 10.1109/LACLO56648.2022.10013456.

[61] W. Li and L. Zhang, "Personalized learning path generation based on network embedding and learning effects," in Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2019, pp. 316–319. doi: 10.1109/ICSESS47205.2019.9040721.

[62] Y. Qiu, "The Impact of LLM Hallucinations on Motor Skill Learning : A Case Study in Badminton," no. July, pp. 139669–139682, 2024.

[63] K. Seo, J. Tang, I. Roll, S. Fels, and D. Yoon, "The impact of artificial intelligence on learner–instructor interaction in online learning," Int. J. Educ. Technol. High. Educ., vol. 18, no. 1, 2021, doi: 10.1186/s41239-021-00292-9.

[64] R. S. Baker and A. Hawn, "Algorithmic Bias in Education," Int. J. Artif. Intell. Educ., pp. 1052–1092, 2022, doi: 10.1007/s40593-021-00285-9.

[65] Y. Zhou, C. Huang, Q. Hu, J. Zhu, and Y. Tang, "Personalized learning full-path recommendation model based on LSTM neural networks," Inf. Sci. (Ny)., vol. 444, no. 55, pp. 135–152, 2018, doi: 10.1016/j.ins.2018.02.053.

[66] S. Amin, M. I. Uddin, A. A. Alarood, W. K. Mashwani, A. Alzahrani, and A. O. Alzahrani, "Smart E-Learning Framework for Personalized Adaptive Learning and Sequential Path Recommendations Using Reinforcement Learning," IEEE Access, vol. 11, no. August, pp. 89769–89790, 2023, doi: 10.1109/ACCESS.2023.3305584.

[67] C. Romero and S. Ventura, "Educational Data mining and Learning Analytics : An updated survey," Univ. Cordoba, 2020.

[68] S. Han, H. Kim, and Y. S. Lee, "Double random forest," Mach. Learn., vol. 109, no. 8, pp. 1569–1586, 2020, doi: 10.1007/s10994-020-05889-1.

[69] Z. Khan et al., "Ensemble of optimal trees, random forest and random projection ensemble classification," Adv. Data Anal. Classif., vol. 14, no. 1, pp. 97–116, 2020, doi: 10.1007/s11634-019-00364-9.

[70] D. McSherry, "Strategic Induction of Decision Trees BT - Research and Development in Expert Systems XV," Res. Dev. Expert Syst. XV, vol. 1, no. Chapter 2, pp. 15–26, 1999, [Online]. Available: http://link.springer.com/10.1007/978-1-4471-0835-1_2%5Cnpapers3://publication/doi/10.1007/978-1-4471-0835-1_2

[71] J. M. Hilbe, "Practical guide to logistic regression," Pract. Guid. to Logist. Regres., vol. 71, no. July, pp. 1–170, 2016, doi: 10.18637/jss.v071.b03.

[72] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," Soft Comput., vol. 25, no. 3, pp. 2277–2293, 2021, doi: 10.1007/s00500-020-05297-6.

[73] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12343 LNCS, pp. 503–515, 2020, doi: 10.1007/978-3-030-62008-0_35.

Appendix

1) Data set sample and amount of data: The experimental dataset contained 500 learner instances, 54 attributes, and 50 sub-field classes. The 54 attributes comprise 50 numerical assessment features, student_id, best_field, best_score, and skill level (Fig. 14).

R15

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	S/No	Name	Gender	Grade	Race	Settlement	Skill Level	Learning Attitude	Learning Motiv	Learning Goal	Learning Interest	Learning Preferenc	Career Goals
2	1	Lim G.	Female	12	Chinese	Urban	Advanced	Collaborative	Intrinsic	Innovation	Mobile Apps	Solitary	Computer Scientist, System Analyst, Researcher
3	2	Abdullah G.	Female	9	Malay	Rural	Beginner	Collaborative	Extrinsic	Innovation	Quantum Comput	Visual	IT Specialist, Systems Analyst
4	3	Martin I.	Male	10	Internatior	Urban	Beginner	Independent	Intrinsic	Innovation	AR/VR	Kinesthetic	Computer Engineer, Embedded Systems Engineer
5	4	Chan P.	Female	12	Chinese	Rural	Beginner	Independent	Extrinsic	Skills	Cloud Computing	Reading/Writing	IS Analyst, Systems Manager
6	5	Naidu A.	Female	12	Indian	Urban	Intermediate	Independent	Intrinsic	Academic	AI & Machine Lear	Kinesthetic	MIS Specialist, IT Manager

W20

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	student_id	best_field	best_score	Skill Level	Computer	Software E	Computer	Informatio	Informatio	Cybersecu	Data Scien	Programm	Mathemati	Logical Re	Problem S	Creativity	Analytical	Communic	Teamwork	Self-Learn	Career Interest	Alignm
2	1	Cybersecu	62	Intermedia	48	56	58	37	39	62	42	4	4	5	5	2	4	3	2	3	3	
3	2	Data Scien	56	Intermedia	29	55	52	45	39	34	56	3	5	5	2	4	5	3	4	2	4	
4	3	Computer	44	Beginner	8	7	44	44	7	25	12	2	4	5	5	4	3	4	3	2	3	
5	4	Software E	67	Intermedia	58	67	41	50	47	54	53	5	3	2	4	5	2	2	4	2	4	
6	5	Computer	39	Beginner	39	16	39	15	27	18	23	5	4	4	4	2	2	4	2	2	2	

Fig. 14. Profile dataset sample.

2) The 50 assessment attributes and best_score are numerical (integer), best_field and Skill Level are categorical, while student_id is an integer identifier excluded from model training.

3) Processing flow: Fig. 15 shows the processing timeline. The assessment attributes were already numerical; therefore, encoding was required only for the categorical prediction label. Fig. 16 shows the label encoding fragment.



Fig. 15. The data processing flow.

The assessment attributes were already numerical; therefore, we needed to encode only the categorical prediction label.

```
X = df.drop(columns=['student_id','best_field',
                     'best_score','Skill Level'])

le = LabelEncoder()
y = le.fit_transform(df['best_field'])
```

Fig. 16. The Feature extraction and target-label encoding code.

4) Choice of models: Random Forest, Decision trees, Logistic regression, and Naïve Bayes: We selected four classifiers to provide complementary classification approaches. We used Random Forest (RF) for its ensemble learning capability, robustness, and ability to model nonlinear relationships [68], [69] while Decision Tree (DT) provides interpretable, rule-based classification [70]. Logistic Regression (LR) was included as a linear probabilistic baseline [71]and Naïve Bayes (NB) as an efficient probabilistic classifier [72]. RF-based feature importance was further used to determine the relative contribution of assessment features to field prediction [73].

5) Testing strategy: The dataset was divided using an 80:20 train-test split, producing 400 training and 100 testing instances; the same split was used for all classifiers to ensure consistent comparison, and performance was evaluated using accuracy, precision, recall, and F1-score.

An Enhanced Deep Learning Framework Integrating Adaptive Whale Optimization and Natural Language Processing for Mental Health Disorder Diagnosis

Nameer N. El.Emam[1], Issa Qabajeh[2]*, Mhammed Almakadmeh[3]
Computer Science Department, College of Information Technology, Amman Arab University, Jordan[1]
Software Engineering Department, College of Information Technology, Amman Arab University, Jordan[2]
École de Technologie Supérieure, Université du Québec, Montréal, Canada[3]

***Abstract*—Mental health disorders are among the biggest global public health challenges, affecting over 1 billion people worldwide and placing a burden on healthcare systems, economies, and societies. Despite progress in clinical psychiatry, mental health professionals still rely heavily on subjective judgments to make diagnoses, leading to delays and inconsistency in diagnosis-making and making early intervention difficult. The evolution of technology-enabled communication channels, i.e., social media, has enabled researchers to employ computational techniques to study digital signatures and patterns and identify psychological distress or behavioral changes in people via their electronic interactions. This study proposes a psychiatric disorder classification system based on Natural Language Processing (NLP) and an Enhanced Deep Learning model powered by the Adaptive Whale Optimization Algorithm (EDL-AWOA). It examines how users in Reddit mental health communities communicate textual data to assist a machine learning algorithm in diagnosing depression and anxiety. It also aims to build a diagnostic support tool using these findings to help identify early signs of mental illness. The results integrate existing measures into a system designed to complement clinical judgment, with the hope that clinicians will adopt these additional steps more quickly. When evaluated with standard metrics of classification accuracy, precision, recall, F1 score, and the confusion matrix, the strong results in the training phase are achieved, including 93.99% accuracy | 94.01% precision | 93.99% recall | 93.99% F1 Score.**

***Keywords*—Psychiatric disorder classification; Natural Language Processing (NLP); Enhanced Deep Learning (E-DL); Adaptive Whale Optimization Algorithm (AWOA); reddit mental health text; depression; anxiety**

I. Introduction

Psychiatric diseases, or mental disorders, are conditions that negatively affect the way you think, feel, and act; they can also hinder a person's daily functioning. They are not just "feelings"; they also shape how a person navigates social connections, whether they succeed or flounder academically or professionally, and their sense of competence in handling day-to-day challenges. In recent years, artificial Intelligence (AI), machine learning (ML), and natural language processing (NLP) have become major resources for healthcare research. Their ability to process massive amounts of unstructured text clinical notes, patient stories, and even social media posts in ways that would be practically impossible by hand is often cited as the reason for this. Over a billion people around the world suffer from mental health disorders, such as anxiety disorders and major depressive disorder (MDD) [1]. Clinical interviews and self-report questionnaires are widely used but limited by time constraints, subjective judgment, cultural issues, and stigma. In recent studies, NLP has been used to identify psychological distress through emotional tone, sentence flow, and word choice in text [2]. Nonetheless, mental health classification is a challenging task as models are easily misled by spoken language variation, do not generalize well, and, of course, suffer from the typical training data bias [3].

To address these drawbacks, researchers have used nature-inspired optimization techniques to better guide machine learning and deep learning models. For parameter tuning, reducing overfitting, and improving classification results, researchers have applied the Whale Optimization Algorithm (WOA) in different variants [4].

Practically, the research focuses on developing an automated system that can assist in early-stage identification of mental health issues. It uses an improved deep learning approach with the Adaptive Whale Optimization Algorithm to create a robust, data-driven system that can help clinicians, psychologists, and mental health organizations detect psychiatric risk by analyzing language use. Tayebi Arasteh et al. [5] conducted research. Wongkoblap et al. [6] proposed that large language models can also facilitate more efficient, automated machine-learning workflows in clinical research. Researchers can leverage the massive amounts of data available on platforms like Reddit to detect real-time, population-level trends in mental illness, helping enable prevention efforts (2022). Researchers have also found that machine-learning models can detect and assess depression and other mental health disorders in people based on their online speech [7].

In theory, the research shows how NLP systems can leverage natural optimization to create better models of human psychological expression through language. It incorporates deep learning and the Adaptive Whale Optimization Algorithm (AWOA), which can improve a model's performance on novel data and higher-order language-independent tasks. The work connects theoretical explanations for how a representation of digital language maps across structures of human psychology to support the utility of NLP in mental health and opportunities

*Corresponding author.

for collaborative work between AI (e.g., [6]; [7]), psychology, and computational behavioral sciences.

Natural Language Processing: Natural Language Processing has come a long way, but diagnosing mental health disorders with clinical accuracy is still an open problem. Current diagnostic methods still rely heavily on clinician interpretation of the patient and their presentation, which may lead to misdiagnosis or delayed treatment. As [8] points out in their 2024 article, text-based NLP models have tried to address this by recognizing patterns in what people write. Nevertheless, they face the same limitations: model interpretability is dicey, datasets are biased, privacy and fairness are still issues, and performance drops off a cliff outside English. Most annotated datasets are aggregated in English, leaving a gap in generalization to other languages, and most models are trained for a single task but cannot perform satisfactorily enough to recognize emotional and linguistic characteristics in posts from people with mental illness. This necessitates systems that classify better, use more attention-based and contextual methods, and generalize to multilingual data.

- How effectively can an enhanced deep learning-based two-hidden-layer neural network model using the adaptive whale optimization algorithm (AWOA) classify psychiatric disorders, such as depression and anxiety, from social media posts?
- Which discourse and lexico-semantic elements in user language are the most crucial indicators for classifying particular mental health disorders?
- How does the proposed EDL-AWOA classification model compare with standard Deep Learning and WOA-based models in terms of classification performance, interpretability, and generalization?

Correspondingly, the objectives are:

- To design and implement a specialized deep learning architecture derived from the Adaptive Whale Optimization Algorithm (AWOA) for accurate classification of depression and anxiety using text data from Reddit.
- To identify and analyze the salient lexico-semantic features and structural elements of discourse in user-generated text that serve as primary indicators for distinguishing among various psychiatric conditions.
- To evaluate the optimized deep learning model against baseline deep learning and WOA-based classification models across accuracy, precision, recall, F1-score, AUC-ROC, interpretability, and generalization ability.

Finally, although the proposed EDL-AWOA framework showed promising performance in identifying mental health conditions from Reddit posts, the study has several limitations. First, the model was developed and evaluated using self-reported social media data rather than clinically verified patient records. Consequently, the system should be considered a computer-aided diagnostic support tool intended to assist with early detection and risk assessment, not to replace professional clinical evaluation. Second, the dataset was limited to English-language Reddit content, which may limit the generalizability of the findings to other languages, cultures, and online platforms. Therefore, future research should validate the proposed framework using clinically annotated datasets and real-world healthcare data in collaboration with mental health professionals to further assess its clinical applicability, robustness, and reliability.

II. Related Work

Several studies have examined machine learning, deep learning, transformer-based models, and various optimization methods for mental health classification using social media text, speech signals, and clinical datasets. In general, these works suggest that computational models can detect linguistic and behavioral cues associated with psychological distress.

Hameed et al. [9] explore the early detection of mental disorders through computational approaches. They used black-box machine learning (ML) models, including Support Vector Machines (SVM), Random Forests (RF), Extreme Gradient Boosting (XGB), and Artificial Neural Networks (ANN). We employed advanced Natural Language Processing (NLP) techniques, including TF-IDF, Latent Dirichlet Allocation (LDA), N-grams, Bag of Words (BoW), and GloVe embeddings, to extract linguistic and semantic features. To address the interpretability limitations of black-box models, Explainable AI (XAI) methods were integrated, specifically Local Interpretable Model-Agnostic Explanations (LIME). However, it was based on English-language datasets and did not use advanced optimization techniques to improve model convergence and generalization.

Senanayake & Liang [10]. Transformers ensemble of DistilBERT, RoBERTa, and XLNet for depression detection in Reddit posts. This late-fusion strategy improved Macro-F1 by 6–8% over individual transformer models, suggesting that combining multiple language representations can be helpful. However, the model was resource-intensive and showed only average performance outside Reddit.

Baydili et al. In [11], a framework that uses the CW-INCA feature-selection algorithm combined with two potent deep learning models, BERT, to identify depressive and suicidal tendencies was proposed. Evaluations on six mental health datasets achieved 99.9% classification accuracy, including the Kaggle Depression Reddit dataset. Although this framework was encouraging, the model required substantial computational resources; it used only English datasets, and it did not use adaptive optimization strategies like AWOA.

Farruque et al. [12] proposed utilizing Mental-BERT along with semi-supervised learning and zero-shot learning methods to profile depression symptoms from social media text. The iterative self-training framework they proposed extends the dataset from 377 to 4,567 samples and improves the Macro-F1 score by 14%. However, several symptom classes remained underrepresented, and the dataset could not be made publicly available due to privacy limitations.

Arif et al. [13] evaluated traditional machine-learning, deep-learning, and transformer-based architectures to classify Reddit posts into six mental-illness categories. Their transformer encoder with late fusion of post titles and content

achieved the best performance, with an F1-score of 89.65%. Nonetheless, it had difficulty differentiating between disorders with the same linguistic manifestations, such as anxiety disorder and bipolar disorder, and its performance was evaluated only using Reddit data.

Nayak et al. [14] Depression Detection via TF–IDF, n-gram Features, SVM, Random Forest and LSTM Models. Their results showed that the LSTM model performed much better than traditional machine-learning methods, reaching around 85% accuracy. However, class imbalance, small dataset size, and a lack of optimization/hybrid learning mechanisms affected the study.

The work, which focused on speech-based depression diagnosis and was presented in 2024, proposed integrating Long Short-Term Memory (LSTM) networks with the Whale Optimization Algorithm (WOA). The model classified the DAIC-WOZ dataset using MFCC features, achieving about 84%. Despite the performance gain from optimization, the framework was evaluated on a single speech dataset, which is computationally expensive.

Hosseinali, F. [15] proposed an improved Selective Backpropagation algorithm (SBP+) to speed up neural network training while improving generalization. The main idea is that during training, the algorithm excludes samples that are already classified correctly with sufficient confidence, rather than applying backpropagation to all training instances at every iteration. This reduces unnecessary weight updates and focuses learning on more informative, difficult samples. However, this study introduces additional hyperparameter-tuning requirements and does not extensively investigate scalability and effectiveness across diverse large-scale application domains. Moreover, the available description does not indicate extensive validation on large-scale NLP, computer vision, or multimodal datasets.

Moosavi et al. [16] suggested using a metaheuristic algorithm, namely the Whale Optimization Algorithm (WOA), to solve constraint optimization problems. The nature-inspired algorithm follows a spiral bubble-net hunting strategy; it does not get stuck in a local minima solution even if the search space is discontinuous.

Scherbakov et al. [17]. An AI-assisted scoping review of 1,768 studies to explore the application of Natural Language Processing (NLP) techniques in mental health research and analyze how different Social Determinants of Health (SDOH) are captured within mental health analytics was conducted. The review found that most studies relied on clinical notes and social media data; few incorporated SDOH factors, confirming a significant gap in the literature. The main limitation is that this investigation is a review, not a new predictive model; other limitations include the limited use of SDOH variables in general practice, the unavailability of datasets for multiple studies described, and a lack of standardized reporting practices across the literature.

In general, past research shows that social-media text contains useful linguistic signals for detecting mental health disorders. While classification performance has been improved by deep learning, various transformer models, and better optimization algorithms in recent years, many issues persist, such as high computational complexity, heavy dependence on the English language, dataset imbalance, limited generalization to other languages or more social science datasets like microblog texts rather than pure text forms like news or articles, etc., as discussed below. These limitations inspire the development of the EDL-AWOA model, where an advanced NLP feature extraction approach is integrated with an adaptive whale optimization technique to improve classification performance, robustness, and generalization in psychiatric disorder detection.

III. Formal Definitions

This section presents the formal mathematical foundations used by the proposed model, covering text feature representation, the deep learning formulation, and the whale optimization mechanism.

Definition 1: Let ID_w be the identity word level that distinguishes between various pieces of writing. The term weighting $TF\_IDF_{t,d}$ of a term t in a document d is defined by the following Eq. (1):

$$ID\_w : TF_{t,d} \times IDF_t \rightarrow TF\_IDF_{t,d} \quad (1)$$

Where the map gives the term-frequency and inverse-document-frequency components in Eq. (2) and Eq. (3):

$$TF_{t,d} = \frac{f_{t,d}}{\sum_{t'} f_{t',d}} \quad (2)$$

$$IDF_t = \log\left(\frac{N}{df_t}\right) \quad (3)$$

where, T is the set of all terms in the corpus; D is the set of all documents; $f(t, d)$ is the number of times term t appears in document d; N is the total number of documents; and $df(t)$ is the number of documents that contain t. $TF\_IDF_{t,d}$ acts as a statistical weighting system that assigns a high value to a term that appears frequently in one document but rarely across the rest of the corpus.

Definition 2: Let the character-level (n_gram) extraction function C of a text string S be defined by the following Eq. (4):

$$n\_gram: s_i \times s_{i+1} \times \cdots \times s_{i+n-1} \rightarrow C_{n,S} \; \forall i \in [1, m - n + 1] \quad (4)$$

where, $\forall i \in [1, m-n+1]$ represents all text strings over the alphabet Σ; m is the string length of S; and $P(\Sigma^n)$ is the set of character sequences of length n. n_gram is created by applying Eq. (1) to the elements of C(n, s) instead of full words, allowing the model to recognize spelling variations, slang, repeated-letter patterns, and partial words.

Definition 3: Let the Latent Semantic Analysis LSA_k transformation be obtained from the truncated Singular Value Decomposition (SVD) of a term-document matrix X. A complete formal definition is typically written as the Eq. (5):

$$LSA_k: R^{|T|\times|D|} \rightarrow R^{k\times|D|} \text{ with } X \approx U_k \Sigma_k V_k^T \quad k \in |T| \quad (5)$$

where, T denotes the set of terms (words) in the vocabulary, D denotes the set of documents, X is the original term–document matrix; U_k and V_k^T contain the reduced left and right singular vectors obtained via Singular Value Decomposition; Σ_k contains the top k singular values; and k is the selected number of latent dimensions. *LSA* is a dimensionality-reduction technique often applied after *TF-IDF* to reduce noise and computational cost while preserving important relationships between documents and terms [18].

Definition 4: Let a layer ℓ of a feed-forward deep learning model be defined by the following Eq. (6) and (7):

$$l:R^n \rightarrow R^m \text{ and } \quad z_l = Wx+b \quad \forall x \in R^n \tag{6}$$

$$a = f(z) \tag{7}$$

where, $x \in \mathbb{R}^n$ is the input vector; $W \in \mathbb{R}^{m \times n}$ is the weight matrix; $b \in \mathbb{R}^m$ is the bias vector; z is the weighted input; $f(.)$ is the activation function; and a is the layer's output activation. A feed-forward deep learning model is a composition of successive layer maps that transmits information from the input layer through multiple hidden layers to the output layer [19].

Definition 5: Let the softmax output function (σ) be defined by the following Eq. (8):

$$\sigma:R^K \rightarrow (0,1)^K \text{ and } \quad \sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \forall i=1,\ldots,K \tag{8}$$

where, $\sigma(z_i)$ is the predicted probability of class i; z_i is the score for class i; K is the total number of classes; and the denominator is the sum of exponential scores over all classes. The softmax function serves as the final layer for classification tasks, transforming output scores into probability values, and the class with the maximum probability is selected as the prediction [20].

Definition 6: Let D be the distance between the current whale location and the top (best) solution defined by the Eq. (9) and Eq. (10)-(12).

$$Distance: F(C, X^*, X_t) \rightarrow D \tag{9}$$

$$\text{where, } F(C, X^*, X_t) = |C \cdot X^* - X_t| \tag{10}$$

and

$$X_{t+1} = X^* - A \cdot D \tag{11}$$

with

$$A = 2ar - a, \quad C = 2r \tag{12}$$

where, X_t is the current solution (whale position) at iteration t; X^* is the best solution found so far; A and C are coefficient vectors; r is a random value in [0, 1]; a is a control parameter that decreases linearly during the search; and t_{max} is the maximum number of iterations. The Whale Optimization Algorithm is a nature-inspired algorithm based on humpback whale hunting behavior. In this algorithm, each whale represents a candidate solution, and the algorithm balances exploration of new regions of the solution space with exploitation toward the best solution [4].

Definition 7: Let the merged labeled (MLable) dataset of D be defined by the union of the two collected Reddit mental health datasets; see the Eq. (13):

$$\text{MLable: } \bigcup_{i=1}^{2} D_i \rightarrow D \tag{13}$$

where, D is defined in the Eq. (14)

$$D = \{(s_i, y_i): s_i \in S_{raw} \text{ and } y_i \in Y\} \quad \forall i=1,\ldots,n \tag{14}$$

where, D_1 and D_2 are the first and second Reddit mental health datasets after column-name standardization; s_i is the raw text of the ith post; y_i is its normalized class label; S_{raw} is the set of raw Reddit posts; Y is the set of diagnostic categories; and n is the total number of samples. Records with missing text or labels are excluded from D.

Definition 8: Let S denote the set of normalized text strings over a given alphabet. Define the preprocessing function P in the Eq. (15):

$$P: REM \times EMJ \times LOW \rightarrow S \tag{15}$$

where, LOW: S → S is the lowercase transformation defined by mapping each character of a string to its lowercase equivalent. EMJ: S → S is the emoji normalization function that replaces each emoji symbol occurring in a string with its corresponding textual description according to a predefined emoji dictionary. REM: S → S is the removal function that deletes URLs, subreddit references. Thus, for any input string s ∈ S: P(s) = REM(EMJ(LOW(s))), where P first converts all characters to lowercase, then replaces emoji symbols with their textual descriptions, and finally removes URLs, subreddit mentions, and user mentions.

Definition 9: Let the tokenization function TOK of a normalized string be defined by the following Eq. (16):

$$\text{TOK: } S \rightarrow W^*, \quad \text{TOK}(s) = (w_1, w_2, \ldots, w_q) \quad \forall s \in S \tag{16}$$

where, W^* denotes the set of finite token sequences, and q is the number of tokens produced from the string s.

Definition 10: Let W be a vocabulary of tokens, let W* denote the set of all finite token sequences over W, and let SW ⊆ W be the set of English stop words. Define the token filtering function in Eq. (17):

$$FILT: W^* \rightarrow W^* \text{ And } FILT(w_1, \ldots, w\_q) = (w_i : w_i \notin SW \text{ and } |w_i| > 1) \tag{17}$$

where, $|w_i|$ denotes the number of characters in token w_i.

Thus, FILT removes all stop words and all tokens of length one while preserving the original relative order of the remaining tokens.

Definition 11: Let the WordNet lemmatization function LEM be defined by the following Eq. (18) to replace each token with its lemma:

$$\text{LEM:W} \rightarrow \Lambda, \quad \text{LEM}(w_1,\ldots,w_q) = (\text{LEM}(w_1),\ldots,\text{LEM}(w_q)) \tag{18}$$

where, $\Lambda \subseteq W$ is the set of WordNet lemmas (canonical dictionary forms), so each inflected token maps to its base form.

Definition 12: Let S_{raw} denote the set of raw text strings and let S_{clean} denote the set of cleaned token sequences in Eq. (19).

$$\text{CLEAN}: S_{raw} \rightarrow S_{clean} \text{ and}$$

$$\text{CLEAN} = \text{REC} \circ \text{LEM} \circ \text{FILT} \circ \text{TOK} \circ \text{NORM} \quad (19)$$

where, REC: W* → S_clean reconstructs a cleaned text sequence from the resulting lemmatized tokens.

Hence, for every raw text string $s \in S_{raw}$, and CLEAN(s) = REC(LEM(FILT(TOK(NORM(s))))).

Thus, CLEAN transforms a raw text string into its fully normalized, tokenized, filtered, lemmatized, and reconstructed cleaned representation.

Definition 13: Let $m \in N$ be the dimension of a hidden-layer activation space, and let $p \in (0, 1)$ denote the dropout rate; the dropout regularization function $DROP_p$ is defined in Eq. (20):

$$DROP_p : R^m \rightarrow R^m \text{ and } \quad DROP_p(a) = \frac{a \odot m}{1-p} \quad (20)$$

$$\text{where, } m = (m_1, \ldots, m_m)$$

$\in \{0,1\}^m$ *and* m_i is a sample of $Bernoulli(1-p) \; \forall \; i = 1, \ldots, m$

Here, $a = (a_1, \ldots, a_m) \in R^m$ is the activation vector of a hidden layer, m is a binary dropout mask, $\odot$ denotes elementwise multiplication, and p is the probability that a neuron is dropped.

The scaling factor $1/(1-p)$ implements inverted dropout, ensuring that $E[DROP_p(a)] = a$, so that the expected activation remains unchanged during training. During training, $DROP_p$ randomly deactivates a fraction p of the neurons to reduce overfitting and improve generalization. During inference, dropout is disabled, and the mapping reduces to the identity function $DROP_p(a) = a$. Thus, $DROP_p$ acts as a stochastic regularization operator that randomly masks hidden-unit activations while preserving their expected magnitude.

Definition 14: Let the categorical cross-entropy loss function L and the gradient-based weight-update map be defined by the following Eq. (21) and (22):

$$L: (0,1)^K \times Y \rightarrow R^+ \text{ and } \quad L(\sigma(z), y) = -\sum_{j=1}^{K} 1[y=j] \log \sigma(z)_j \quad (21)$$

where, $\sigma(z)$ is the softmax output of the proposed deep learning (see *Definition 5)*; $y \in Y$ is the true class label; $1[\cdot]$ is the indicator function.

Furthermore, the cross-entropy loss function L is applied to update the weight (w_{ij}) connecting neuron i to neuron j, using the learning rate (α) and the gradient of the loss function $\nabla L(\cdot)$ for the previous weight [21]. The training process is based on mini-batch backpropagation, Eq. (20), to calculate the new weight w_{ij}^{new}.

$$w_{ij}^{new} = w_{ij}^{old} - \alpha \nabla L\left(w_{ij}^{old}\right) \quad \forall i,j \quad (22)$$

IV. The Proposed EDL-AWOA Model

A. EDL-AWOA Architecture

The proposed EDL-AWOA model uses a hybrid machine learning setup that pairs deep learning with an adaptive whale optimization method. It begins with raw data collection, then moves to preprocessing for data cleaning and transformation. The processed data feeds into an enhanced deep learning design with an input layer, several hidden layers, and an output layer that produces initial predictions. At the same time, an adaptive whale optimizer improves the model by tuning parameters through adaptive search and refinement strategies inspired by whale behavior. The interaction between the deep learning model and the optimizer improves learning performance, and the final outputs are passed to the classification and evaluation stage, where the model's predictive performance is assessed. Fig. 1 shows this full workflow.

The Enhanced MLP identifies psychiatric conditions from linguistic representations of Reddit posts. The input layer processes TF-IDF, char-TF-IDF, or LSA feature vectors. The model uses two fully connected hidden layers, each with ReLU activation followed by dropout regularization. Two layers were chosen to keep the design simple and effective while still learning more complex nonlinear patterns from the extracted text features. Dropout was added to reduce overfitting, helping the system generalize better to unseen Reddit posts. The output layer applies a softmax function and produces a probability distribution over six target classes: Anxiety/Stress, Bipolar, Major Depressive, Normal, Personality Disorder, and Substance Abuse. In addition, Fig. 1 shows the enhanced deep learning architecture.

For feature extraction, the study uses four techniques: word-level TF-IDF, character-level TF-IDF, a combination of word- and character-level TF-IDF, and Latent Semantic Analysis with TF-IDF using Truncated SVD for dimensionality reduction. Dimensionality reduction is applied in specific situations to reduce operational costs and maintain proper memory performance in deep learning.

The Adaptive Whale Optimization Algorithm (AWOA) tunes the main hyperparameters of the enhanced deep learning model. In this implementation, the optimized hyperparameters are the learning rate and the dropout rate, and each whale agent represents a candidate hyperparameter configuration. The learning rate is searched in the range $[10^{-4}, 10^{-2}]$, and the dropout rate in the range [0.10, 0.50]. AWOA improves the standard Whale Optimization Algorithm by combining opposition-based learning, Lévy flight exploration, adaptive inertia weight, adaptive spiral movement, and an epoch-switched alpha mechanism. The epoch-switched alpha mechanism changes the alpha value within the same optimization run based on the training epoch phase: the first alpha value applies during the early training phase, before the switch epoch, while the second alpha value applies during the later training phase, after the switch epoch. For example, in the schedule 0.7 ↔ 0.3, the optimizer uses 0.7 before the switch epoch and 0.3 afterward. The reverse schedule, 0.3 ↔ 0.7, is evaluated separately to study how the order of alpha values

affects exploration–exploitation behavior and classification performance.

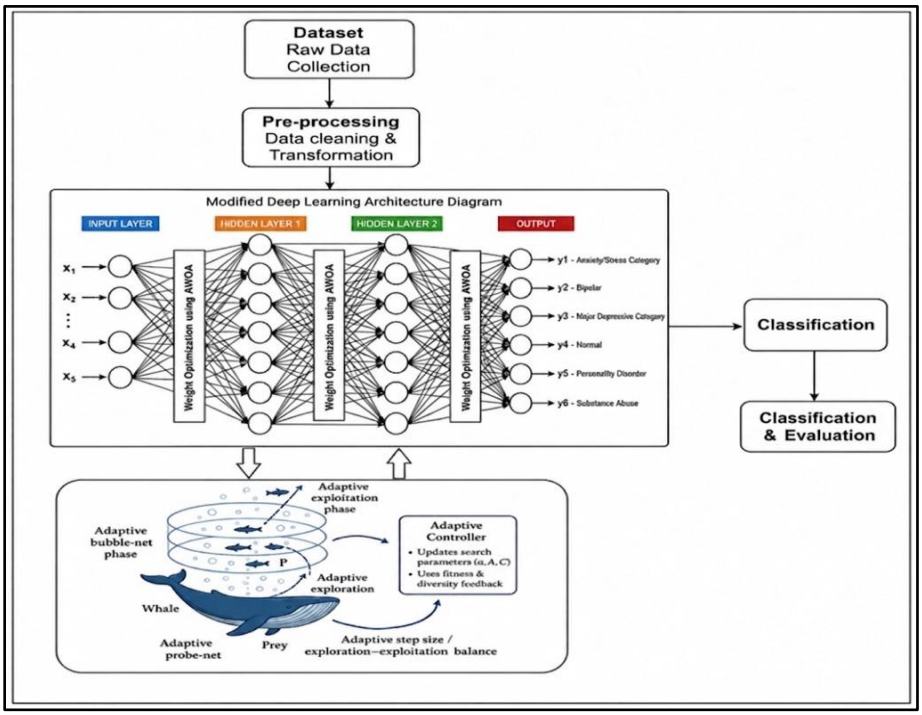


Fig. 1. The architecture of the EDL-AWOA model.

B. Algorithms of the Proposed Model

The methodology is implemented using three algorithms: a data pre-processing algorithm (Algorithm 1), the enhanced deep learning algorithm (Algorithm 2), and the adaptive whale optimization algorithm (Algorithm 3).

Algorithm 1: Data Pre-Processing Algorithm

Input: Raw Reddit text dataset

Output: Cleaned text dataset

Procedure:

Step 1: Load and merge the collected datasets. *(see Definition 7)*

- Step 1.1: Load the first Reddit mental health dataset
- Step 1.2: Load the second Reddit mental health dataset.
- Step 1.3: Standardize the column names in both datasets, such as text and label.
- Step 1.4: Append the rows of the second dataset to the first dataset.
- Step 1.5: Create one unified dataset containing all Reddit text samples and their corresponding labels.

Step 2: Normalize class labels. *(see Definition 7)*

Step 3: Remove records with missing text or labels. *(see Definition 7)*

Step 4: For each text sample: *(see Definition 8)*

- Step 4.1: Convert text to lowercase. *(see Definition 8, map LOW)*
- Step 4.2: Convert emojis to textual descriptions. *(see Definition 8)*
- Step 4.3: Remove URLs, Reddit mentions, and user mentions. *(see Definition 8)*

Step 5: Tokenize the cleaned text. *(see Definition 9)*

Step 6: Remove stop words and one-character tokens. *(see Definition 10)*

Step 7: Apply WordNet lemmatization. *(see Definition 11)*

Step 8: Reconstruct the cleaned text. *(see Definition 12)*

Step 9: Remove posts with fewer than three words. *(see Definition 12)*

Step 10: Remove duplicate posts. *(see Definition 12)*

Step 11: Return the cleaned dataset. *(see Definition 12)*

Step 12: End.

Algorithm 1 integrates two Reddit mental health datasets into a unified corpus by standardizing column names, merging records, normalizing class labels, and removing instances with missing values to ensure data consistency and quality. It then applies a sequence of text-cleaning operations, including lowercasing, emoji-to-text conversion, removal of URLs and user mentions, tokenization, stop-word filtering, and WordNet lemmatization, thereby reducing noise and standardizing linguistic variations. Finally, the cleaned tokens are reconstructed into text, while very short posts and duplicate entries are discarded, resulting in a high-quality dataset suitable for reliable machine learning and natural language processing tasks.

We first merged the Reddit datasets after standardizing class labels and column names, then removed records with missing values. Text preprocessing included lowercasing, emoji-to-text conversion, removal of URLs, subreddit references, and user mentions, followed by tokenization, stop-word removal, elimination of single-character tokens, and WordNet lemmatization. We extracted features using word-level TF-IDF (n-gram range = 1–2, maximum vocabulary = 20,000), character-level TF-IDF (n-gram range = 3–5,

maximum vocabulary = 10,000), and LSA with 300 latent dimensions via Truncated SVD. The E-DL model employed two hidden layers with 128 and 64 neurons using ReLU activation, while the dropout rate (0.10–0.50) and learning rate (10^{-4}–10^{-2}) were optimized by AWOA; training was conducted with a batch size of 32, 32 epochs, an AWOA population size of 30 whales, and 15 optimization iterations.

Algorithm 2: Enhanced Deep Learning Model (E-DL)

Input: TF-IDF feature vectors; Class labels

Output: Predicted class; Evaluation results

Procedure:

Step 1: Load the pre-processed dataset from Algorithm 1. *(see Definition 12)*

Step 2: Convert the text data into TF-IDF feature vectors. *(see Definition 1-3)*

Step 3: Define the Enhanced DL architecture *(see Definition 4)*

Step 3.1: Define the input layer according to the TF-IDF feature size. *(see Definition 4, input $x \in \mathbb{R}^d$)*

Step 3.2: Define the first fully connected dense layer with 128 neurons and ReLU activation. *(see Definition 4)*

Step 3.3: Apply dropout regularization. *(see Definition 13)*

Step 3.4: Define the second fully connected dense layer with 64 neurons and ReLU activation. *(see Definition 4)*

Step 3.5: Apply dropout regularization. *(see Definition 13)*

Step 3.6: Define the softmax output layer for class prediction. *(see Definition 5)*

Step 4: Define the loss function and evaluation metric. *(see Definitions 14 and 16)*

Step 5: Train the E-DL using mini-batch learning and backpropagation. *(see Definition 14)*

Step 6: Optimize the E-DL hyperparameters using AWOA, as described in Algorithm 3, until the maximum number of epochs is reached. *(see Definition 6)*

Step 7: Evaluate the trained model using accuracy, precision, recall, F1-score, and AUC. *(see Definition 16)*

Step 8: Save the final results.

Step 9: End.

The proposed Enhanced Deep Learning (E-DL) algorithm transforms the preprocessed Reddit text into TF-IDF feature vectors and processes them through a multi-layer neural network with two fully connected, ReLU-activated dense layers (128 and 64 neurons) and dropout regularization to improve feature learning and prevent overfitting. The model is trained using mini-batch backpropagation, while the Adaptive Whale Optimization Algorithm (AWOA) optimizes hyperparameters to improve classification performance. Finally, the trained E-DL model predicts mental health categories and is evaluated using comprehensive metrics, including accuracy, precision, recall, F1-score, and AUC, to measure its effectiveness and robustness.

Algorithm 3: Adaptive Whale Optimization Algorithm (AWOA)

Input: deep learning architecture defined in Algorithm 2 and in Fig. 1.

Set the Initial value of the learning rate search space $\alpha \in [10^{-4}, 10^{-2}]$

Where 10^{-4} is the lower bound and 10^{-2} is the upper bound.

Output**:** Find the best α

Procedure

Step 1: Define the initial set of weight functions w_{ij} in deep learning. *//see Definition 4*

Step 2: Set the initial whale position (X_0) and set the learning rate (α) as in Eq. (23) *//(see Definition 6)*

$$\alpha = X_0 \tag{23}$$

Step 3: For $s = 0$ *to* $N - 1$.

where, N - 1 is the last whale position

Step 3.1: Calculate the new Whale Position X_{s+1} using Eq. (24) *//see Definition 6*

$$X_{s+1} = X_s - \bar{F}(X_s, X_p) \tag{24}$$

where, X_s is the sth position of whales and X_p is the prey location, and $\bar{F}(.)$ is the whale optimization model [22].

Step 3.2: Apply the Adaptive WOA approach based on the fitness function $Fit(.)$ to calculate the best learning rate α as in Eq. (25) and Eq. (26):

$$\alpha = \beta X_{s+1} + (1 - \beta) X_s \tag{25}$$

where

$$\beta = \begin{cases} 0.9, & \text{if } \mathrm{Fit}(X_{s+1}) > \mathrm{Fit}(X_s) \\ 0.1, & \text{if } \mathrm{Fit}(X_{s+1}) \leq \mathrm{Fit}(X_s) \end{cases} \tag{26}$$

Step 3.3: Calculate the new value of weight w_{ij}^{new} using Eq. 22) *//see Definition 14,*

Step 3.4: Apply the training process of deep learning using an activation function between every two layers using Eq. (27): *(see Definition 4, maps 6–7)*

$$z = f\left(b + \Sigma_i \Sigma_j a_{i,j} \times w_{i,j}^{new}\right) \tag{27}$$

where, $f(.)$ is the ReLU activation function, and b is the DL bias.

Step 4: End // *Find the best α of DL*

The Adaptive Whale Optimization Algorithm (AWOA) is integrated with the Enhanced Deep Learning (E-DL) model to dynamically optimize the learning rate during training by simulating whale-hunting behavior. At each iteration, the WOA search mechanism updates whale positions, and a fitness function evaluates candidate learning rates to select the value that minimizes model loss and improves convergence. The selected learning rate then updates network weights via backpropagation, while ReLU activation functions enable effective feature learning across layers, yielding an optimized deep learning model with improved classification performance.

The AWOA algorithm optimizes classification accuracy on the validation dataset, defined as (Validation accuracy, Validation F1-score, Validation Macro AUC). We repeated each optimization experiment more than 20 times to minimize stochastic effects and recorded the average performance. We fixed the best hyperparameter configuration from the validation set and evaluated it on the independent test set without further optimization to prevent data leakage.

V. Results and Discussion

This chapter discusses the experimental outcomes from implementing the proposed EDL-AWOA model. The main goal is to test how well the proposed model can classify psychiatric disorders using Reddit mental health text data, with

a special focus on depression and anxiety. We evaluate the results using common classification metrics, including accuracy, precision, recall, F1-score, macro AUC, and confusion matrix analysis, and compare them with baseline machine learning and deep learning models to assess the contribution of the AWOA optimization process.

A. System Setup

The research was conducted on a Colab server with 64GB of RAM, running the G4 runtime with an NVIDIA RTX PRO 6000 GPU and approximately 96GB of VRAM, with CUDA 13.0-accelerated training. Python 3.11 (managed via pyenv) was the primary programming language. PyTorch was used for model building and training; Scikit-learn for preprocessing, feature extraction, evaluation, and basic machine learning methods; NLTK for natural language preprocessing; and imbalanced-learn for handling class imbalance. NumPy and Pandas were used for data analysis, and Matplotlib for visualizing model performance. The system supports repeatable experiments by automatically logging model configurations, confusion matrix outputs, hyperparameter settings, and the final trained model weights.

The dataset is the public Kaggle Reddit Mental Health Dataset, which contains posts from mental health-related subreddits focused on depression and anxiety. We retained only English-language posts with legitimate self-reported mental health expressions. We split the dataset into 70% training, 15% validation, and 15% testing sets, and Table I shows the training and optimization configuration.

TABLE I. TRAINING AND OPTIMIZATION PARAMETERS

Parameter	Value/Condition
Batch size	32
Learning rate	0.001 default; optimized from 0.0001 to 0.01
Epochs	32
Pre-training Epochs	4
Noise dimension	100
Hidden layer 1	Dense, 128 neurons, ReLU
Hidden layer 2	Dense, 64 neurons, ReLU
Hyperparameters optimized	Learning rate, Dropout
AWOA – Population Size	30
AWOA – Max Iterations	15

The batch size was set to 32 to balance training stability and computational efficiency. The default learning rate was fixed at 0.001 for steadier convergence but was set within the AWOA search bounds of 0.0001 to 0.01. The number of training epochs was set to 32 to give the model enough opportunity to learn meaningful patterns without excessive compute time. Two dense hidden layers were used: the first with 128 neurons and ReLU activation, and the second with 64 neurons and ReLU activation. We optimized the learning rate and dropout, since both strongly affect performance. For AWOA, the population size was set to 30 whale agents and the maximum number of iterations to 15, providing a workable balance between optimization quality and computational cost.

B. Experiments on the Proposed Model to Find the Best Parameter Tuning

Let the evaluation metrics be defined by the following maps (28–32) from the confusion-matrix counts of the test set:

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \quad (28)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (29)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (30)$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (31)$$

$$\text{AUC} = \int_0^1 TPR \, d(FPR) \quad (32)$$

where, TP, TN, FP, and FN are the true-positive, true-negative, false-positive, and false-negative counts, respectively; TPR = Recall is the true-positive rate; and FPR = FP / (FP + TN) is the false-positive rate. For the multi-class setting, the metrics are macro-averaged over the K diagnostic categories.

The proposed model was evaluated across different conditional alpha (α) schedules during training. The results show that model performance varied according to the selected α setting, indicating that the adaptive coefficient directly affects the optimization process and classification performance. Among the tested settings, Experiment 7, using the α schedule 0.9 ↔ 0.1, achieved the best performance, with 93.99% accuracy, 94.01% precision, 93.99% recall, 93.99% F1-score, and 0.9993 macro AUC. Table II presents the full results.

TABLE II. EDL-AWOA PERFORMANCE ACROSS ALPHA SETTINGS (TRAINING)

Exp.	Adaptive α Selection	Accuracy	Precision	Recall	F1-Score	Macro AUC	Adaptive α Selection	Accuracy
1	0.7 ↔ 0.3	87.20%	87.22%	87.20%	87.19%	0.9968	1	0.7 ↔ 0.3
2	0.3 ↔ 0.7	85.80%	85.81%	85.80%	85.79%	0.9959	2	0.3 ↔ 0.7
3	0.8 ↔ 0.2	89.51%	89.56%	89.51%	89.51%	0.9978	3	0.8 ↔ 0.2
4	0.2 ↔ 0.8	86.58%	86.64%	86.58%	86.59%	0.9964	4	0.2 ↔ 0.8
5	0.6 ↔ 0.4	88.40%	88.43%	88.40%	88.40%	0.9973	5	0.6 ↔ 0.4
6	0.4 ↔ 0.6	87.61%	87.62%	87.61%	87.61%	0.9969	6	0.4 ↔ 0.6
7	**0.9 ↔ 0.1**	**93.99%**	**94.01%**	**93.99%**	**93.99%**	**0.9993**	**7**	**0.9 ↔ 0.1**
8	0.1 ↔ 0.9	86.21%	86.23%	86.21%	86.21%	0.9962	8	0.1 ↔ 0.9

The accuracy was between 85.80% and 93.99% in the eight experiments. Experiment 7 achieved the highest accuracy using the 0.9 ↔ 0.1 schedule, indicating that the best balance between exploration and exploitation came from starting with a larger alpha value and selecting lower alpha values after the switch epoch. Similarly, the schedule from step 0.3 ↔ 0.7 (the lowest accuracy) achieved 85.80% accuracy.

C. Confusion Matrix and AUC-ROC Curves for EDL-AWOA Model

The confusion matrix corresponding to the best EDL-AWOA configuration is shown in Fig. 2.

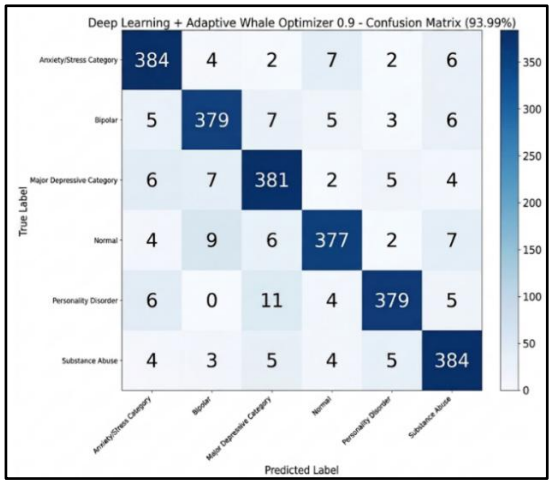


Fig. 2. Confusion matrix for the EDL-AWOA model.

The confusion matrix shows excellent classification performance with an overall accuracy of 93.99%, as indicated by the strong diagonal values (377-384 correct predictions per class) and relatively few misclassifications. Most errors occur between clinically similar classes, particularly Personality Disorder and Major Depressive Category (11 cases), and Normal and Bipolar (9 cases), suggesting some overlap in the learned features. Overall, the Deep Learning model optimized with the Adaptive Whale Optimizer (0.9) demonstrates high reliability and balanced performance across all six classes, with no class exhibiting substantially higher error rates than the others.

Fig. 3 shows the AUC-ROC curves across the six mental health categories shown in Fig. 2.

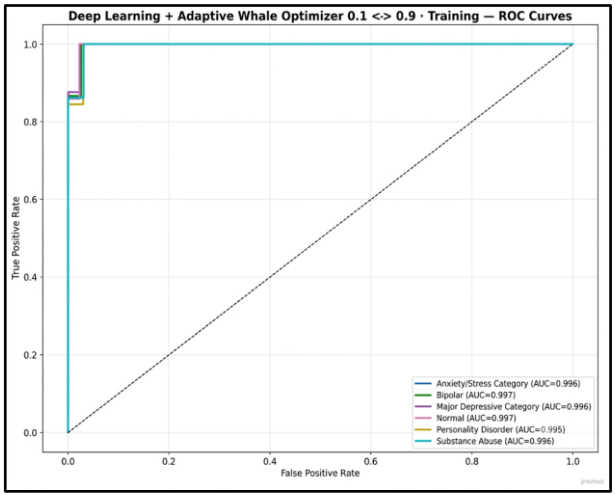


Fig. 3. AUC-ROC curves for the best EDL-AWOA model.

The AUC-ROC curves of the best EDL-AWOA model demonstrate excellent discriminative capability across all six mental health categories, with AUC values ranging from 0.995 to 0.997. The Bipolar and Normal classes achieved the highest AUC of 0.997, while Anxiety/Stress, Major Depressive Category, and Substance Abuse attained 0.996, and Personality Disorder reached 0.995. The ROC curves lie very close to the upper-left corner, indicating a high true positive rate and a very low false positive rate. These results confirm that the EDL-AWOA model provides highly accurate and robust class separation, making it highly effective for multi-class mental health disorder classification.

D. Comparison Between Various Types of Deep Learning

The proposed model (EDL-AWOA) is compared with two baselines: standard deep learning (DL) and deep learning with the standard Whale Optimization Algorithm (DL-WOA).

Fig. 4 shows the accuracy and loss curves for the training, validation, and testing phases of the Deep Learning (DL) model.

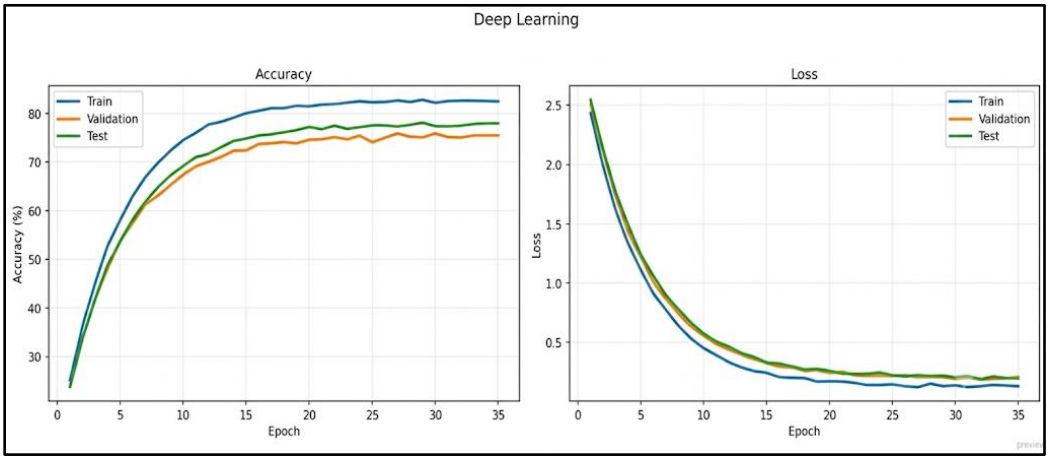


Fig. 4. Training, validation, and testing accuracy/loss curves for the Deep Learning (DL) model.

The deep learning model showed substantial progress, with training accuracy rising from about 25% to 83% (+232%), validation accuracy from 24% to 75% (+213%), and test accuracy from 24% to 78% (+225%) over 35 epochs. At the same time, the loss decreased markedly, with training loss dropping from about 2.45 to 0.12 (95.1% reduction), validation loss from 2.50 to 0.20 (92.0% reduction), and test loss from 2.55 to 0.21 (91.8% reduction). These results indicate effective model convergence and strong generalization, as the validation and test curves closely follow the training curves with only a small performance gap.

Fig. 5 shows the accuracy and loss curves for the training, validation, and testing phases of the Deep Learning (DL) with WOA (DL-WOA) model.

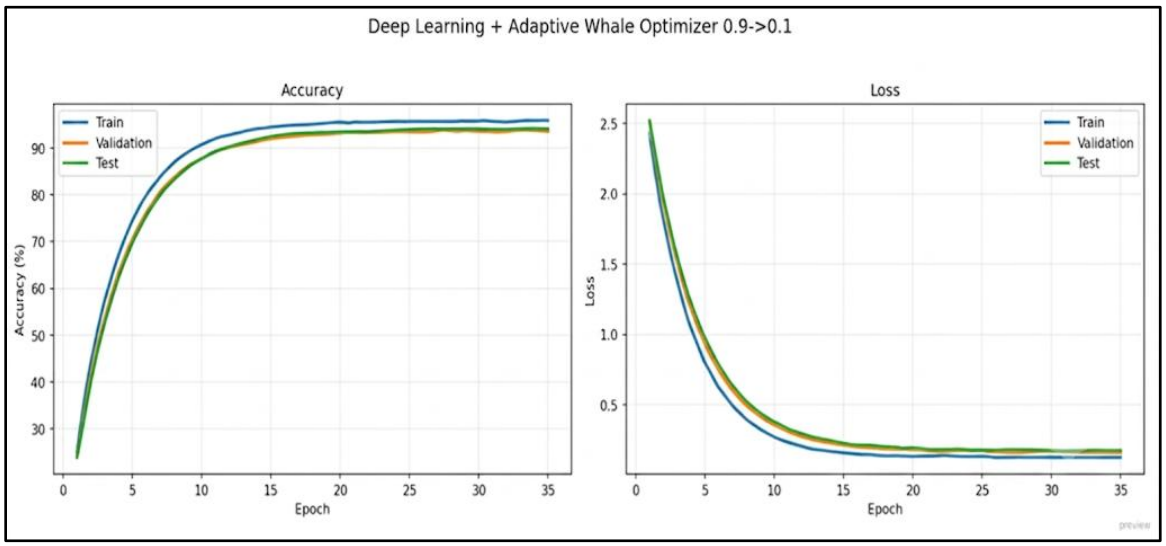


Fig. 5. Training, validation, and testing accuracy/loss curves for the Enhanced Deep Learning and adaptive WOA (EDL-WOA) model.

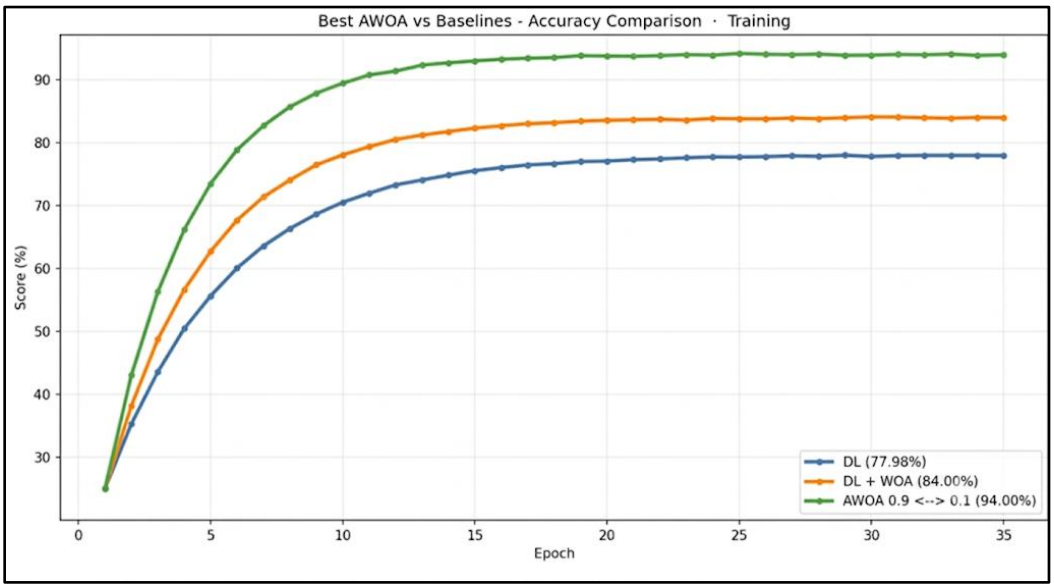


Fig. 6. Accuracy comparison between DL, DL-WOA, and EDL-AWOA (training).

The Deep Learning + Whale Optimizer model outperformed the baseline model, with training, validation, and test accuracies rising from approximately 25% to 87% (+248%), 24% to 82% (+242%), and 24% to 85% (+254%), respectively. Simultaneously, training, validation, and test losses fell from about 2.45, 2.50, and 2.55 to 0.13, 0.18, and 0.20, representing reductions of approximately 94.7%, 92.8%, and 92.2%, respectively, indicating faster convergence, improved generalization, and better predictive performance compared with the standard deep learning model.

Fig. 6 shows the training, validation, and test curves for the best EDL-AWOA configuration using the 0.9 ↔ 0.1 alpha schedule. The accuracy curves increase rapidly during the early epochs and then stabilize after approximately epoch 15, while the loss curves decrease sharply and remain low across training, validation, and test data. The small gap between the curves indicates stable generalization and no major overfitting.

The accuracy and loss curves in Fig. 8 indicate that the proposed deep learning integrated with Adaptive Whale

Optimization (EDL-AWOA) converges quickly and achieves stable performance on the training, validation, and test datasets. Accuracy rises quickly from around 25% (epoch 1) to 95%, 94%, and 93.5% on the training, validation, and test sets, respectively. At the same time, the loss drops from 2.5 to 0.15 on training, 2.5 to 0.18 on validation, and 2.5 to 0.17 on testing, a decrease of around 92%- 94%. The closely aligned training, validation, and test curves indicate strong generalization with little overfitting. Overall, the results show that AWOA effectively optimizes the learning process by speeding convergence, improving classification accuracy, and significantly reducing prediction error.

The accuracy comparison on the training and testing sets is shown in Fig. 7 and 8, respectively.

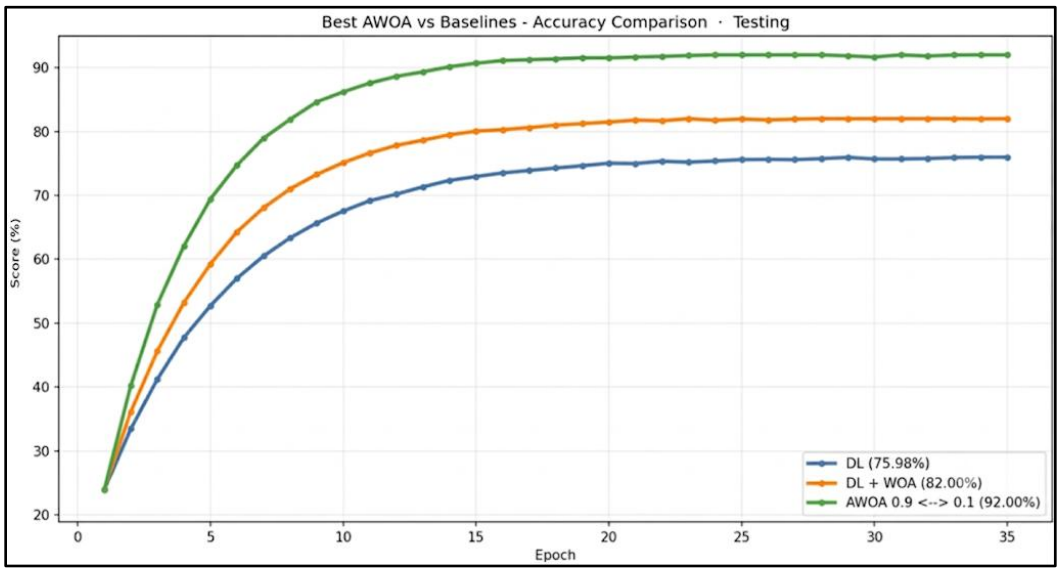


Fig. 7. Accuracy comparison between DL, DL-WOA, and EDL-AWOA (testing).

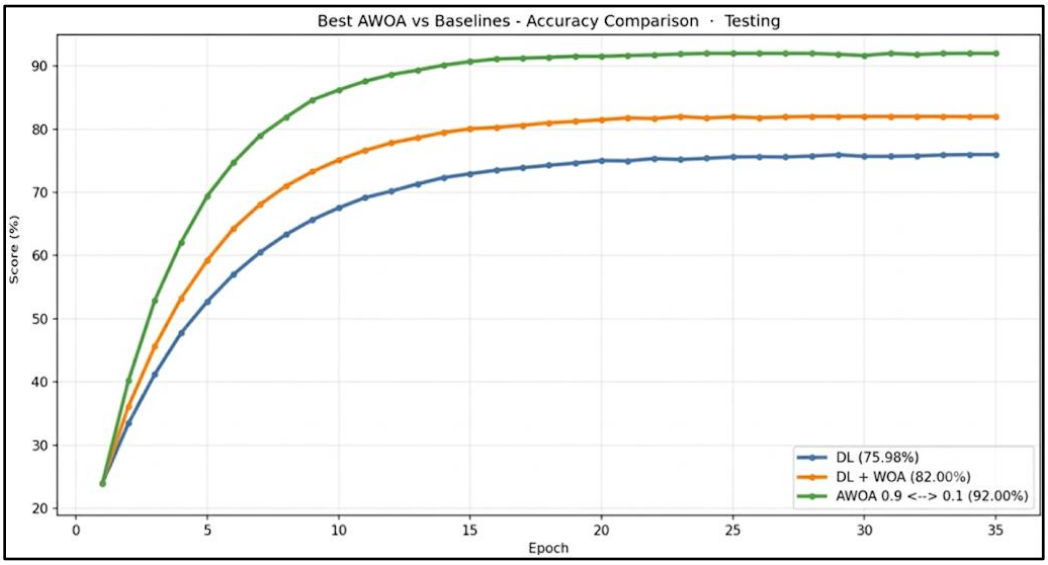


Fig. 8. Accuracy comparison between DL and DL-AWOA (testing).

The training results indicate that the proposed EDL-AWOA model outperforms both conventional DL and DL-WOA. The deep learning model (baseline DL) reached a training accuracy of 77.98%, and incorporating the Whale Optimization Algorithm (DL-WOA) yielded 84%, a 7.72% improvement over DL. The proposed EDL-AWOA model further improved training accuracy to 94.00%, an 11.90% improvement over DL-WOA and a substantial 20.54% improvement over the baseline DL model. In addition, EDL-AWOA converged faster and maintained the highest accuracy throughout training, highlighting the effectiveness of the adaptive optimization mechanism in improving parameter tuning, feature learning, and overall classification performance.

The testing results confirm that the proposed EDL-AWOA model significantly outperforms both the baseline DL and the DL-WOA models in classification accuracy. The conventional DL model achieved a testing accuracy of 75.98%, while incorporating the Whale Optimization Algorithm increased accuracy to 82.00%, a 7.92% improvement over DL. The

proposed EDL-AWOA model further improved testing accuracy to 92.00%, an additional 12.20% improvement over DL-WOA and a remarkable 21.08% improvement over the baseline DL model. Furthermore, EDL-AWOA converged faster and maintained consistently higher accuracy across all testing epochs, showing that the adaptive optimization strategy improves model generalization, parameter tuning, and predictive performance on unseen data.

Tables III and IV compare the proposed model (EDL-AWOA) with two baselines on the metrics (Accuracy, Precision, Recall, F1-score, and AUC) for training and testing, respectively.

TABLE III. COMPARISON OF THE PROPOSED MODEL WITH BASELINE MODELS (TRAINING)

Model	Accuracy	Precision	Recall	F1-Score	Macro AUC
DL	77.98%	77.98%	77.98%	77.98%	0.9903
DL-WOA	84%	84.05%	84%	83.97%	0.9949
EDL-AWOA (Proposed)	94%	94.01%	94%	94%	0.9993

As evaluated against the ground truth, Table III indicates that the performance of each evaluation metric remains strong on the training side, with the highest scores by EDL-AWOA. For accuracy, EDL-AWOA was 94.00%, which is better than both baseline DL (77.98%) and DL-WOA (84.00%), with improvements of 20.54% and 11.90%, respectively. The same trend is also observed with respect to precision, in which case EDL-AWOA had a performance of 94.01%, while the corresponding DL and DL-WOA were 77.98% and 84.05%. Moreover, the recall and F1-score values also improved from 77.98%(DL), approximately 84.00% (DL-WOA), to 94.00% with EDL-AWOA, respectively, which demonstrate the superior classification consistency and balanced performance of the EDL-AWOA method. Moreover, it obtained the highest Macro AUC (0.9993) among all experimental models, including DL (0.9903) and DL-WOA (0.9949), indicating that the classification characteristics of mental disorders were clearly distinguished by the proposed model from each other. The results indicate that integrating the Adaptive Whale Optimization Algorithm could effectively improve performance in feature extraction, parameter tuning, and classification accuracy compared to its non-AWOM counterparts.

TABLE IV. COMPARISON OF THE PROPOSED MODEL WITH BASELINE MODELS (TESTING)

Model	Accuracy	Precision	Recall	F1-Score	Macro AUC
DL	75.98%	75.98%	75.98%	75.98%	0.9853
DL-WOA	82%	82.05%	82%	81.97%	0.9899
EDL-AWOA (Proposed)	92%	92.01%	92%	92%	0.9990

Table IV clearly shows the testing performance of the baseline DL and DL-WOA models against the proposed model (EDL-AWOA), with EDL-AWOA greatly outperforming them on every evaluation metric. The accuracy of the conventional DL model was 75.98%, while with the Whale Optimization Algorithm, it improved to 82.00%, i.e., a gain of 7.92% over DL. The proposed EDL-AWOA model significantly improved the accuracy to 92.00%, a relative increase of 12.20% over DL-WOA and an astounding 21.08% over the baseline DL model, respectively. As shown in Table III, a similar performance improvement is observed in precision, recall, and F1-score; EGL-AWOA obtained an accuracy of about 92.00%, while DL was about 75.98% and DL-WOA was approximately 82.00%, which indicates that EGL-AWOA has better predictive reliability than other methods with better class balance. Additionally, EDL-AWOA achieved the highest Macro AUC of 0.9990 in this test, again higher than DL (0.9853) and DL-WOA (0.9899), demonstrating its superior ability to classify mental health categories with one-vs-all comparisons. These results indicate that the proposed adaptive whale optimization strategy improves model generalization and classification performance on unseen test data.

E. Comparison with Previous Works

Table V compares the proposed EDL-AWOA model with recent studies evaluated on the same Kaggle Reddit mental health dataset. Using a common corpus minimizes the influence of dataset-specific factors and provides a more reliable assessment of relative model performance. The results demonstrate that EDL-AWOA achieves superior classification performance compared with existing approaches evaluated under the same data conditions.

TABLE V. COMPARISON OF THE PROPOSED MODEL WITH RELATED STUDIES

Study	Method	Dataset	ACC.	Prec.	Recall	F1	AUC
Utsa et al. [23].	Interpretable linguistic features (author-disjoint)	RMHD	89%	93%	85%	89%	0.9500
Jena et al. [24].	Continual Neuro-Symbolic Graph Learning (MH-Freeze)		92%	N/R	N/R	92%	0.9420
Kim et al. [25].	Random Forest (Text + LIWC), binary depression		91%	96%	90%	93%	N/R
Proposed Model	EDL-AWOA		93.99%	94.01%	93.99%	93.99%	0.9993

The comparative study shows that the EDL-AWOA model achieved the best performance among RMHD-based studies on mental health classification. Utsa et al. (2026) on the RMHD dataset achieved 89% accuracy and an AUC of 0.9500, and Jena et al. (2026) achieved 92% accuracy and a 92% F1-score, while the proposed model performed better, with maximum accuracy (93.99%), recall (93.99%), and AUC (0.9993). Compared with Kim et al., a Random Forest classifier with text and LIWC features achieved 91% accuracy and a 93% F1-score, while the EDL-AWOA improves accuracy by c.a. 3.29% and the F1-score by 1.06%. Moreover, the exceptionally high AUC of 0.9993 indicates near-perfect class discrimination, substantially exceeding AUC values reported in previous studies. These results confirm that integrating deep learning with the Adaptive Whale Optimization Algorithm enhances feature learning and parameter optimization, leading to superior predictive performance and robustness compared with existing approaches.

VI. Conclusion and Recommendations

This study presents an enhanced deep learning model combined with the Adaptive Whale Optimization Algorithm (EDL-AWOA) for psychiatric disorder classification using Reddit mental health text data. A comprehensive Natural Language Processing (NLP) data preprocessing followed by term frequency–inverse document frequency (TF-IDF)-based feature extraction, a two-hidden-layer deep learning structure, and automatically adaptive hyperparameters after multiple iterations were integrated to effectively predict six mental health categories, including Anxiety/Stress, Bipolar, Major Depressive with or without psychotic features, Normal, Personality Disorder, and Substance Abuse or not. The experimental results indicated that the adaptive optimization strategy can optimize the learning procedure and provide better classification accuracy, faster convergence time, and better generalisation performance.

In the set of analysed alpha schedules, the AWOA configuration (0.9 ↔ 0.1 adaptive) outperformed all others, with an accuracy value of 93.99%, a precision value of 94.01%, and a recall value of 93.99% for the micro and weighted average F1-score values, whilst obtaining results equal in Macro AUC to nearly one at 0.9993 Confusion matrix analysis indicated high classification consistency across all classes, with only a small number of instances in which clinically related disorders were misclassified. Additionally, the ROC analysis confirmed a high discriminative ability (all AUCs were found to need development) for all classes—ranging from 0.995 to 0.997.

Experimental results revealed that EDL-AWOA drastically improved both the DL model and the DL-WOA model in terms of overall performance. The proposed model outperforms DL and DL-WOA by 21.08% and 12.20% (accuracy), respectively, whilst achieving higher scores in precision, recall, F1-score, and Macro AUC in the test dataset. Moreover, the comparison between recent state-of-the-art studies in the field showed that EDL-AWOA obtained the best overall accuracy and AUC values, indicating that adaptive whale-based optimization is a practical approach to effectively solving these mental health classification problems.

The findings confirm that combining adaptive metaheuristic optimization with deep learning yields a robust, adaptable framework with potential for automated psychiatric disorder detection. Thus, the proposed system will make this process more efficient by supporting real-time decision-making for mental health care workers, since they can relatively quickly detect psychopathologies from large volumes of online text data. Nonetheless, future work may explore multilingual datasets, transformer-based embeddings, explainable artificial intelligence methods, and multimodal data sources (speech, clinical records, and behavioral signals) to optimize diagnostic performance and potential utility in the clinic.

An important research direction for the future is to extend the model output from the proposed work on multilingual mental health text classification. Note: This research only applies to English-language Reddit posts, so it does not cover mental health literature in all languages. Future studies should incorporate Arabic and more languages to allow for translation of the model in different ethical and cultural situations since somatic accounts for communicating psychological distress are typically influenced by lexicon, culture, dialect, and social class.

Acknowledgment

The authors express their sincere gratitude to Amman Arab University and the University of Québec for providing the academic environment and research support necessary to conduct this study. The authors also acknowledge the developers and maintainers of the publicly available Reddit mental health datasets and the open-source software libraries used in this research, including Python, PyTorch, Scikit-learn, NLTK, Pandas, and NumPy. Finally, the authors thank the reviewers and editors for their valuable comments and constructive suggestions, which have significantly improved the manuscript's quality, clarity, and rigor.

References

[1] Stein, D. J., Shoptaw, S. J., Vigo, D. V., Lund, C., Cuijpers, P., Bantjes, J., Sartorius, N., & Maj, M. (2022). Psychiatric diagnosis and treatment in the 21st century: Paradigm shifts versus incremental integration. World Psychiatry, 21(3), 393-414. https://doi.org/10.1002/wps.20998

[2] O'Dea, B., Batterham, P. J., Braund, T. A., Chakouch, C., Larsen, M. E., Berk, M., Torok, M., Christensen, H., & Glozier, N. (2025). A randomised cross-over trial examining the linguistic markers of depression and anxiety in symptomatic adults. NPJ Mental Health Research, 4(1), 30. https://doi.org/10.1038/s44184-025-00111-8

[3] Sun, J., Dong, Q.-X., Wang, S.-W., Zheng, Y.-B., Liu, X.-X., & Lu, L. (2023). Artificial intelligence in psychiatry research, diagnosis, and therapy. Asian Journal of Psychiatry, 87, 103705. https://doi.org/10.1016/j.ajp.2023.103705.

[4] Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S., & Faris, H. (2023). A systematic review of the Whale Optimization Algorithm: Theoretical foundation, improvements, and hybridizations. Archives of Computational Methods in Engineering, 30, 4113-4159. https://doi.org/10.1007/s11831-022-09859-7

[5] Arasteh, S. T., Han, T., Lotfinia, M., Kuhl, C., Kather, J. N., Truhn, D., & Nebelung, S. (2024). Large language models streamline automated machine learning for clinical studies. Nature Communications, 15(1), 1603. https://doi.org/10.1038/s41467-024-45943-0

[6] Wongkoblap, A., Vadillo, M. A., & Curcin, V. (2021). Social media big data analysis for mental health research. In Big Data Analytics for Healthcare (pp. 109-143). Elsevier.

[7] Liu, D., Feng, X. L., Ahmed, F., Shahid, M., & Guo, J. (2022). Detecting and measuring depression on social media using a machine learning approach: Systematic review. JMIR Mental Health, 9(3), e27244. https://doi.org/10.2196/27244

[8] Teferra, B. G., Rueda, A., Pang, H., Valenzano, R., Samavi, R., Krishnan, S., & Bhat, V. (2024). Screening for depression using natural language processing: Literature review. Interactive Journal of Medical Research, 13, e55067. https://doi.org/10.2196/55067

[9] Hameed, S., Nauman, M., Akhtar, N., Fayyaz, M. B., & Nawaz, R. (2025). Explainable AI-driven depression detection from social media using natural language processing and black box machine learning models. Frontiers in Artificial Intelligence, 8, 1627078. https://doi.org/10.3389/frai.2025.1627078

[10] Senanayake, A., & Liang, Z. (2025). Anxiety detection in Reddit posts through emotion dynamics analysis. In Proceedings of SCITEPRESS (pp. 920-927).

[11] Baydili, İ., Tasci, B., & Tasci, G. (2025). Deep learning-based detection of depression and suicidal tendencies in social media data with feature selection. Behavioral Sciences, 15(3), 352. https://doi.org/10.3390/bs15030352

[12] Farruque, N., Goebel, R., Sivapalan, S., & Zaïane, O. R. (2024). Depression symptoms modelling from social media text: An LLM-driven semi-supervised learning approach. Language Resources and Evaluation, 58(3), 1013-1041. https://doi.org/10.1007/s10579-023-09683-4

[13] Arif, M., Ameer, I., Bölücü, N., Sidorov, G., Gelbukh, A., & Elangovan, V. (2024). Mental illness classification on social media texts using deep learning and transfer learning. Computación y Sistemas, 28(2). DOI not available.

[14] Nayak, S., Mahapatra, D., Chatterjee, R., Parida, S., & Dash, S. R. (2022). A machine learning approach to analyze mental health from Reddit posts. In Smart Innovation, Systems and Technologies (pp. 357-366). DOI not available.

[15] Hosseinali, F. (2024). Accelerated gradient descent using improved selective backpropagation. Expert Systems with Applications, 255, 124426. https://doi.org/10.1016/j.eswa.2024.124426

[16] Moosavi, S. K. R., Akhter, M. N., Zafar, M. H., & Mansoor, M. (2024). Constraint optimization: Solving engineering design problems using Whale Optimization Algorithm (WOA). In S. Mirjalili (Ed.), Handbook of Whale Optimization Algorithm (pp. 193-216). Academic Press. https://doi.org/10.1016/B978-0-32-395365-8.00021-X

[17] Scherbakov, D. A., Talwar, A., Desai, P., Roche, K. L., Kwan, B. M., & Iturralde, E. (2025). Natural language processing and social determinants of health in mental health research: Scoping review. JMIR Mental Health, 12, e67192. https://doi.org/10.2196/67192

[18] Qi, Q., Hessen, D., Deoskar, T., & Van der Heijden, P. G. M. (2024). A comparison of latent semantic analysis and correspondence analysis of document-term matrices. Natural Language Engineering, 30(4), 722-752. https://doi.org/10.1017/S1351324923000289

[19] Sajid, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2024). Neuro-fuzzy random vector functional link neural network for classification and regression problems. IEEE Transactions on Fuzzy Systems, 32(5), 2738-2749. https://doi.org/10.1109/TFUZZ.2023.3330012

[20] Rashedi, K. A., Nejatian, S., Parvin, H., Rezaie, V., & Minaei-Bidgoli, B. (2024). Multi-layer perceptron-based classification with missing data. Journal of Risk and Financial Management, 17(2), 69. https://doi.org/10.3390/jrfm17020069

[21] Reyad, M., Sarhan, A. M., & Arafa, M. (2023). An enhanced Adam algorithm for deep neural network optimization. Neural Computing and Applications, 35(23), 17095-17112.

[22] Reddy, K., & Saha, A. K. (2022). An enhanced Whale Optimization Algorithm for exploitation capability and stability enhancement. Heliyon, 8(10), e11019. https://doi.org/10.1016/j.heliyon.2022.e11019

[23] Utsa, A. D. (2026). Early linguistic pattern of anxiety from social media using interpretable linguistic features: A multi-faceted validation study with author-disjoint evaluation (arXiv:2601.11758). arXiv. https://arxiv.org/abs/2601.11758

[24] Jena, M., Khan, N., Lee, M. Y., & Rho, S. (2026). Neuro-symbolic graph learning for causal inference and continual learning in mental-health risk assessment. Computer Modeling in Engineering & Sciences. DOI https://doi.org/10.32604/cmes.2025.075119.

[25] Kim, S., Imieye, O., & Yin, Y. (2025). Interpretable depression detection from social media text using LLM-derived embeddings (arXiv:2506.06616). arXiv. https://arxiv.org/abs/2506.06616

MedViT-MT: A Multi-Task Medical Vision Transformer for Glioblastoma Brain Analysis with Temporal Encoding and Clinical Data Fusion

Sana Cheema[1], Ghulam Gilanie[2], Muhammad Faheem Mushtaq[3], Rania Aboalela[4], Ali Daud[5]
Department of Artificial Intelligence, Islamia University, Bahawalpur, Punjab 63100, Pakistan[1, 2]
Department of Data Science, The Islamia University of Bahawalpur, Bahawalpur, Punjab 63100, Pakistan[3]
Department of Information Systems-Faculty of Computing and Information Technology,
King Abdulaziz University, Rabigh, Saudi Arabia[4]
Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates[5]

***Abstract*—Glioblastoma Multiforme (GBM) is the most invasive form of primary brain tumor. The average survival time after diagnosis is only 15 months. Accurately and automatically analyzing GBM MRI scans remains a major clinical challenge. It requires the tumor to be localized, a report to be generated in a structured way, and information from multiple data sources to be integrated simultaneously. Existing methods address these problems separately and use models for one task that do not leverage the complementary information available across segmentation, captioning, and grounding. This study introduces MedViT-MT (Medical Vision Transformer Multi-Task), a unified transformer architecture that performs tumor segmentation, molecular-subtype classification, lesion grounding, and patient-conditioned structured caption generation from GBM DSC-MRI sequences. MedViT-MT employs Swin Transformer Tiny as its hierarchical visual transformer backbone for extracting multiscale imaging features, followed by the addition of a clinical fusion module that is based on cross-attention and that directly places tabular clinical features (IDH1, MGMT, KPS, GTR, and GTR) into the visual feature space. It further extracts inter-slice volumetric information from the MRI slice sequence, helping the model infer three-dimensional tumor volume. On the UPENN-GBM dataset, which contains 375 patients with DSC MRI sequences and their clinical records, MedViT-MT reaches 98.31% classification accuracy, 99.37% AUC-ROC, 91% mean Dice score, and 100% few-shot accuracy (1-shot, 5-shot, 10-shot). The model is also able to generalize well to different IDH1 subtypes, and it runs using 16GB RAM on a CPU. The separate experiments to identify what each architectural component delivers show that all of them matter; the statistical analyses indicate the reliability of the model.**

Keywords*—Glioblastoma; multi-task learning; deep learning; transformer; clinical data fusion; temporal encoding

I. Introduction

Brain tumors are one of the most severe neurological conditions that have a profound impact on the survival and quality of life of patients [1]. To support diagnosis, therapy planning, and evaluation of prognosis, it is vital to perform accurate tumor detection, precise tumor localization, and detailed characterization of tumor subregions [2]. Magnetic Resonance Imaging (MRI) is currently the most widely used imaging technique for assessing brain tumors, as it provides excellent visualization of soft-tissue structures and supports multiple sequences [3, 4]. The use of artificial intelligence for automated brain tumor analysis has become a significant research focus driven by the increasing accessibility of multi-modal MRI datasets from sources such as The Cancer Imaging Archive and the establishment of the Brain Tumor Segmentation Challenge as a benchmark [5].

Brain tumors are usually examined through MRI scans of different sequences, such as T1, T1 contrast-enhanced (T1c), T2, and FLAIR, to identify tumor parts like the enhancing tumor, necrotic core, and edema. In the clinic, radiologists manually check these sequences to draw lesion boundaries, calculate tumor burden, and characterize radiological features through narrative reports [6]. This method takes up a lot of time, and different people may disagree on the results. As the amount of imaging data keeps increasing and there is a demand for precision medicine, automated systems that can help clinicians in tumor interpretation and decision-making are becoming necessary [7].

The clinical examination and evaluation of brain tumors nowadays is being done mainly through radiologists examining multi-sequence MRI visually using manual or semi-automatic tumor segmentation tools, grading radiologically based on morphological features to perform histopathological examination after biopsy or surgery, and integrating imaging results with clinical and molecular data. These techniques are still considered the gold standard in neuro-oncology and have a crucial role in confirming diagnosis and planning therapy [8]. However, the traditional clinical methods still have several limitations despite their great importance. Manual cut-out of the work from images is tiring and subject to differences in interpretation of the results by different experts [9]. Looking at surfaces visually may miss patterns that are located deeply in data obtained from multiple sequences. Obtaining histopathological confirmation means doing highly intrusive procedures, which cannot always be done. Clinical workflows as they stand nowadays usually generate narrative-based reports rather than reports with quantitative uncertainty estimates, which essentially means that the confidence level can only be assessed based on someone's judgment or perception. All of this

points to the necessity of dependable, automated, and interpretable computational support tools [10].

During the last several years, there has been a significant growth trend in the use of machine learning and deep learning techniques for the automated detection and segmentation of brain tumors. Convolutional Neural Networks (CNNs) such as the U-Net and its different versions have been very successful in volumetric tumor segmentation tasks [11]. Transformer-based architectures have recently taken medical imaging to a new level by enhancing the understanding of global context [12]. Classification of tumor types and prediction of survival outcomes using MRI features is also among the areas that have been extensively studied [13]. Medical image captioning is another area that is gaining attention, and it is focused on generating radiology-type descriptions from imaging data [14].

Although the majority of deep learning methods promise great results, some major issues are raised by most existing ones [15]. A large number of the models do not go beyond segmentation and classification and hence miss out on structured clinical information. The models designed for caption generation usually depend on global image features, which makes them incapable of relating textual descriptions solely to the tumor area. Besides, they usually don't take into account uncertainty estimation, which means they give overconfident predictions that may cause a loss of clinical trust [16]. Thus, at present, these tools are more like isolated predictive instruments rather than comprehensive, clinically meaningful decision-support frameworks[17].

Despite the huge steps made in brain tumor segmentation and classification, there are still quite a few gaps. First, there is not much research on single frameworks that can do segmentation, classification, and radiology-style report generation all at the same time [18]. Second, most of the vision-language methods have not been updated to include segmentation-guided grounding, so it is not guaranteed that the texts produced by these methods describe the tumor areas correctly. Third, the lack of structured clinical metadata in the model design means the potential for patient-specific reasoning is not being realized. Lastly, the importance of uncertainty-aware modeling for dependable clinical use is hardly recognized in multi-task brain tumor systems[19].

The main contributions of this research are outlined below:

- This research proposes a unified multi-task deep learning framework that simultaneously performs brain tumor segmentation, molecular-subtype classification, lesion grounding, and patient-conditioned structured caption generation, enabling comprehensive GBM analysis within a single model.
- This study introduces a segmentation-guided visual grounding mechanism that generates tumor-aware region tokens and aligns visual representations with generated structured clinical captions, thereby improving the spatial interpretability of the textual output.
- This research develops a cross-attention transformer-based fusion module to combine the structured clinical metadata, which allows the model to incorporate patient-specific context and support more informed patient-aware decision-making.
- This study incorporates uncertainty-aware modeling to generate confidence maps with the predictions to enhance the interpretability, reliability, and trustworthiness of the system in clinical applications.

The rest of the study is structured as follows. Section II discusses the literature on brain tumor segmentation, multi-modal fusion, and medical report generation. Section III details the proposed methodology used to cover the segmentation branch, clinical fusion mechanism, uncertainty modeling, and caption generation module. Section IV is for the presentation of the quantitative results, ablation studies, and comparisons with state-of-the-art methods. Section V: Conclusion and Future Work.

II. Literature Review

Deep learning has made significant advancements in brain tumor segmentation and classification using MRI data. However, combining clinical data, uncertainty modeling, and report generation into a single integrated system is still quite limited.

A. Brain Tumor Segmentation in Multi-Modal MRI

A multimodal transformer-based architecture was introduced for brain tumor segmentation using incomplete MRI data [20]. The approach combines a modality-specific encoder, a multi-modal transformer, and a shared-weight decoder to generate missing modalities. The experiments on BraTS 2018, 2019, and 2020 datasets show that the model outperforms the state-of-the-art models in terms of segmentation accuracy, especially when modalities are missing. This study [21] presents a deep learning-based method for brain tumor segmentation and classification. The authors used DeepLabV3+ integrated with Atrous Spatial Pyramid Pooling (ASPP) and Bayesian Optimization. The main features of the work are a couple of custom-made models, namely, IRB-96 and IRB-Self, which are designed for feature extraction and classification, respectively. The features obtained from the model are optimized through Bayesian methods, and to make the results more understandable, LIME is used. Their method reaches 92.68% segmentation accuracy and 95.42% classification accuracy on the dataset, which is better than other advanced techniques. Recently, a novel technique, namely Contextual Transformer U-Net (CT-UNet), was introduced for brain tumor segmentation[22]. It is a hybrid of Transformer blocks and U-Net architecture that effectively outlines tumors in multimodal MRI sequences. On the BraTS dataset, CT-UNet achieved a Dice coefficient of 0.92, sensitivity of 0.90, and specificity of 0.94, therefore outperforming the conventional models. Furthermore, it showed very accurate spatial localization with a Hausdorff distance of 7.5 mm. Besides, the proposed method enables rapid and precise segmentation, which in turn allows clinical application for improving patient care and treatment planning. Multi-Modal Segment Anything Model (MSAM) for glioma tumor segmentation is proposed by integrating structural MRI with diffusion tensor imaging (DTI) data [23]. The approach feature first a multi-modal feature fusion block, missing modality training technique to handle missing data

scenarios. MSAM on the BraTS 2018 dataset outperforms U-Net not only in terms of Dice Similarity Coefficient but also in 95% Hausdorff Distance, particularly when only structural MRI data is available. These results support the ability of MSAM to discard a modality and still yield reliable segmentation, even with incomplete data or varying tumor sizes.

B. Uncertainty-Aware Segmentation for Trustworthy AI

A deep learning approach to brain tumor segmentation was proposed that combines uncertainty estimation with an attention-based multimodal fusion mechanism [24]. The presented system is based on a multi-encoder 3D U-Net architecture combined with Bayesian uncertainty estimation for accurate segmentation. The numerical comparisons on the BraTS 2018 and BraTS 2019 datasets demonstrate that their approach outperforms the state-of-the-art methods in Dice score, Hausdorff distance, and sensitivity. This method significantly improves segmentation accuracy at the tumor border, and it is flexible to different network architectures and computer vision tasks. A brain tumor segmentation method is introduced by incorporating multi-modal fusion and uncertainty quantification techniques [25]. Fuchs et al. [26] explored the feasibility of uncertainty measures in brain cancer delineation and put forward a multi-headed Variational U-Net (mu-VNet). This approach is based on the variational inference principle merged with the ensemble's variability features to produce better uncertainty assessments without sacrificing much on the computational requirement. They tested this model on the BraTS 2018 data in the in-distribution set-up and four distribution-shift scenarios. It is noteworthy that instead of the teacher-student multimodal fusion described in [25], the main novelty of [26] is reliable uncertainty estimation, calibration techniques, and resistance to out-of-distribution inputs. Such characteristics make this work especially helpful to engineers and scientists who are designing accurate systems for medical image segmentation. The FLAIR and T1c modalities instruct the T2 and T1 modalities in a teacher-student fusion manner. The model employs Monte Carlo dropout for the uncertainty calculation and introduces an uncertainty-aware loss function. On the BraTS 2018, 2019, and 2020 datasets, the method scored 86.6% Dice Similarity Coefficient for whole tumor, 87.1% for tumor core, and 79.0% for enhancing tumor, while the Hausdorff Distance was only 4.2 mm, surpassing the state-of-the-art methods.

C. Radiology Report Generation and Vision–Language Modeling

MediVLM is a proposed that is the transformer-based vision-language model that combines visual encodings and text representations through cross-attention to generate structured clinical captions from medical images [27]. Furthermore, it is capable of functioning with very limited paired training data by relying on object detection and cross-modal alignment. Another approach to automatic radiology report generation that takes into account chest X-rays [28]. It relies on basically two modules: 1) a learned knowledge base that extracts and stores medical knowledge automatically from textual embeddings, and 2) multi-modal alignment that is used to ensure semantic congruence between X-ray images, reports, and disease labels. The proposed method was evaluated on the public IU-Xray and MIMIC-CXR datasets by means of both Natural Language Generation (NLG) and clinical efficacy (CE) metrics. The findings indicate that their model surpasses the performance of state-of-the-art methods in different metrics; thus, it proves the effectiveness of the learned knowledge base and multimodal alignment. The assessment reveals that clinical accuracy has been considerably improved, with a significant increase in precision, recall, and F1 scores, particularly in clinical efficacy metrics. Medical Vision Attention Generation (MedVAG) model is a reliable system for the automatic generation of medical reports [29]. The main idea of the model is to make use of Vision Transformer (ViT)-based visual feature extraction and combine it with GPT-2 for language modeling. The model is further improved through graph-based feature fusion and multiple attention mechanisms (co-attention, cross-attention, and memory-guided attention) that are aimed at ensuring the consistency and accuracy of the medical diagnosis. Tested on the IU X-Ray and COV-CTR datasets, MedVAG was able to outperform state-of-the-art methods in terms of several metrics, namely BLEU, METEOR, ROUGE, and CIDEr, as well as in clinical effectiveness measures. Through ablation experiments, it was found that the use of attention mechanisms and feature fusion plays a crucial role in the association of visual and textual information. MedVAG can be highly effective as a radiologist's assistant, helping to ease the workload and increase the accuracy of diagnoses.

This research identifies four critical gaps in existing literature: 1) no unified model jointly addresses segmentation, captioning, and grounding for GBM; 2) clinical tabular data is rarely fused with imaging features in a principled cross-attention framework; 3) the temporal sequential structure of DSC MRI slices is seldom exploited; and 4) few-shot and zero-shot learning modes for GBM classification remain unexplored. This work addresses all four gaps simultaneously.

III. The Proposed Methodology

In this section, the complete methodology of the proposed Medical Vision Transformer Multi-Task (MedViT-MT) framework is presented, based on a deep learning unified system designed for glioblastoma multiforme (GBM) analysis using DSC-MRI sequences along with clinical metadata. The framework combines segmentation, classification, visual grounding, and patient-conditioned structured clinical caption generation into one comprehensive (end-to-end trainable) model. Unlike regular single-task models, MedViT-MT is able, via hour-by-hour cross-modal fusion and temporal reasoning, to simultaneously capture both spatial and volumetric features of tumors shown in Fig. 1.

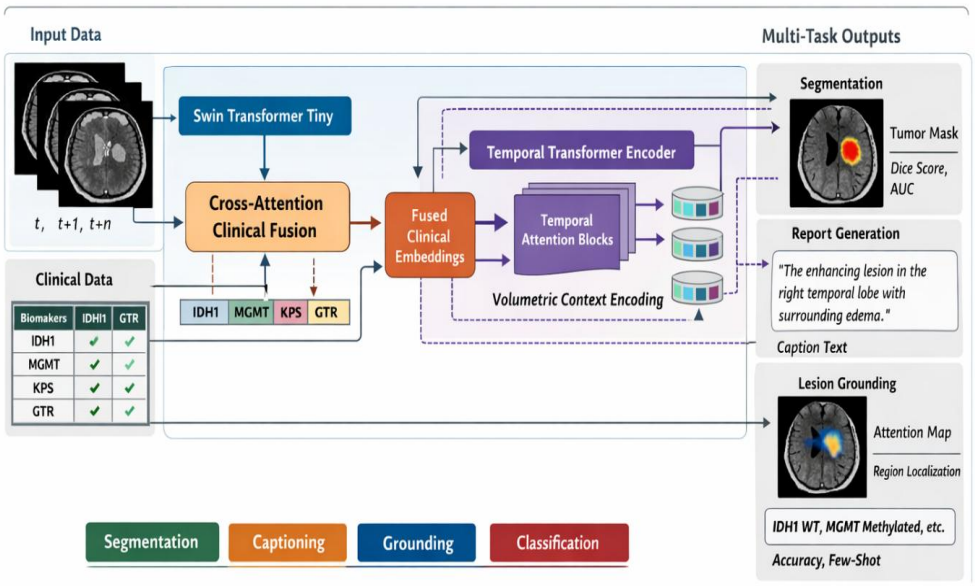


Fig. 1. MedViT-MT proposed architecture.

The input to the model consists of a sequence of MRI slices and a corresponding vector of structured clinical features. Let

$$X = \{x_1, x_2, \ldots \ldots x_T\} \quad x_T \in R^{3\times H\times W} \tag{1}$$

Denote the sequence of T DSC MRI slices, where H=W=224H. The associated clinical vector is defined as

$$c \in R^{D_c}, D_c = 7 \tag{2}$$

To optimize the four tasks segmentation, caption generation, grounding, and classification simultaneously, the model formulates a unified objective function as

$$\tau_{total} = \tau_{seg}\,\tau_{seg} + \tau_{cap}\,\tau_{cap} + \tau_{gnd}\,\tau_{gnd} + \tau_{cls}\,\tau_{cls} \tag{3}$$

where, each λ controls the contribution of its respective task. This formulation enables joint learning and cross-task regularization, improving generalization performance across all outputs.

The framework consists of four successive steps. Initially, a hierarchical transformer backbone is employed by the model to extract discriminative visual features from a series of MRI slices. Secondly, the clinical fusion module that merges the patient's structured information with the image features is used to enrich the visual features. Thirdly, a temporal transformer encoder is utilized to capture the relationships among the slices, resulting in the production of context-aware representations along the entire MRI sequence. Lastly, the representations that have been learned are forwarded to the task-specific heads, which will be responsible for producing tumor masks, lesion localization outputs, descriptive reports, and subtype predictions (Algorithm 1). The main idea behind this architecture is that the model will be able to combine pixel-level comprehension, region-level localization, patient-level classification, and language-level clinical summarization into one unified learning system.

Algorithm 1: MedViT-MT Forward Pass

```
INPUT: image_seq [B, T, 3, 224, 224], clinical [B, 7], valid_mask [B, T]
OUTPUT: seg_masks, cap_logits, boxes, scores, cls_logits
-- STAGE 1: Backbone Feature Extraction --
B, T, C, H, W = image_seq.shape
x = image_seq.reshape(B*T, C, H, W)        # Flatten batch+time
img_feats = SwinBackbone(x)                # [B*T, 768]
img_feats = LayerNorm(img_feats)
img_feats = Dropout(img_feats)
img_feats = img_feats.reshape(B, T, 768)   # Restore time dim
-- STAGE 2: Clinical Cross-Attention Fusion --
clin_proj = MLP_clinical(clinical)         # [B, 256]
clin_exp  = clin_proj.unsqueeze(1).expand(B,T,256) # [B,T,256]
img_proj  = Linear_img(img_feats)          # [B, T, 256]
cross_attn_out, _ = MultiheadCrossAttn(Q=img_proj, K=clin_exp, V=clin_exp)
fused = LayerNorm(img_proj + Dropout(cross_attn_out))
sa_out, _ = MultiheadSelfAttn(fused, fused, fused)
fused = LayerNorm(fused + Dropout(sa_out))
fused = LayerNorm(fused + Dropout(FFN(fused))) # [B, T, 256]
-- STAGE 3: Temporal Encoder --
cls_token = learnable_cls.expand(B, 1, 256)
x_temp = concat([cls_token, fused], dim=1)  # [B, T+1, 256]
x_temp = x_temp + positional_embedding[:T+1]
x_temp = TransformerEncoder(x_temp, key_padding_mask=~valid_mask)
x_temp = LayerNorm(x_temp)
cls_out = x_temp[:, 0, :]                  # [B, 256] Global token
seq_out = x_temp[:, 1:, :]                 # [B, T, 256] Per-slice
-- STAGE 4: Task Heads --
seg_masks = SegmentationHead(seq_out)      # [B, T, 2, 224, 224]
cap_logits = CaptioningHead(cls_out, tgt_tokens) # [B, 64, 1000]
boxes, scores = GroundingHead(seq_out)     # [B,T,10,4], [B,T,10,2]
cls_logits = ClassificationHead(cls_out)   # [B, 3]
-- LOSS COMPUTATION --
L = w_seg*L_seg + w_cap*L_cap + w_gnd*L_gnd + w_cls*L_cls
RETURN seg_masks, cap_logits, boxes, scores, cls_logits
```

A. Dataset Description

UPENN-GBM, which is a dataset comprising multi-parametric MRI scans from patients diagnosed with Glioblastoma Multiforme (GBM) at the University of Pennsylvania Health System, is accessible publicly through The Cancer Imaging Archive (TCIA). This dataset consists of de-identified images. Here, we used UPENN-GBM as a base dataset since it offers clinically meaningful brain tumor imaging data and enables multimodal analyses like tumor characterization, subtype prediction, and integrated medical AI modeling.

1) MRI data: The imaging portion of the dataset includes Dynamic Susceptibility Contrast (DSC) MRI scans performed before the operation. DSC-MRI is a way to measure brain activity by following a contrast agent temporarily through brain vessels and is used for estimating cerebral blood volume and hemodynamics. In the case of glioblastoma, this type of imaging can provide valuable insights into tumor-related factors such as blood vessel formation, leakage of the blood-brain barrier, and microvascular structure anomalies, which are tumor aggressiveness and subtype characterization-related aspects.

Each patient's MRI data for this research is arranged as a series of two-dimensional PNG image files placed in patient-specific directories following the UPENN-GBM naming style, for example, UPENN-GBM-00001_11. Protocol and anatomical coverage differences usually result in exposure of roughly 10 to 50 slices per patient. As the MedViT-MT framework that we propose models the sequences to train a batch effectively, it is inevitable to have a fixed length of the input. Thus, the maximum sequence length that can be handled is set at T = 10 slices/patient to keep the computation within reach of the hardware having 16 GB RAM. When there are more slices than T, the middle ones are taken since the central part of the scan usually showcases the tumor most clearly. This approach to selecting slices conserves visibility of the lesion while giving a standard input size to all patients.

2) Clinical data: Besides the imaging data, the UPENN-GBM dataset comprises a structured clinical info file provided in comma-separated value format, namely UPENN-GBM_clinical_info_v2_1.csv. It contains 12 clinical variables linked to each patient and is utilized to construct the multimodal fusion part of the proposed design. These variables are demographic characteristics, molecular markers, survival-related details, and treatment-associated indicators; separately or altogether with the image features, they provide patient-specific context that can enhance decision-making for the images and ultimately classification performance.

Among these variables are patient identifier, sex, age at scan, time of survival after surgery, survival status, censoring data, IDH1 mutation status, MGMT methylation status, Karnofsky Performance Scale (KPS), gross total removal status, time elapsed since baseline pre-operative imaging, and pseudoprogression/true progression score. While some of these variables, e.g., IDH1 and MGMT, are directly related to tumor characteristics and prognosis, others like KPS and gross total resection status add functional and treatment-related context. Since missing data is present to different extents in each variable, great attention must be paid during preprocessing so that missing values do not cause inconsistencies in model training. The clinical variables together with their estimated percentages of missing values are listed in Table I.

TABLE I. CLINICAL VARIABLES IN THE UPENN-GBM DATASET

Variable	Type	Description	Missing (%)
ID	String	Patient identifier (e.g., UPENN-GBM-00001_11)	0%
Gender	Categorical	Patient sex: Male (M) / Female (F)	0%
Age_at_scan_years	Continuous	Patient age at time of MRI scan	0%
Survival_from_surgery_days	Continuous	Survival duration from surgical resection	~5%
Survival_Status	Categorical	Deceased / Alive / Lost to Follow-up	0%
Survival_Censor	Binary	Censoring indicator for survival analysis	~60%
IDH1	Categorical	IDH1 mutation: Wildtype / Mutated / NOS/NEC	0%
MGMT	Categorical	MGMT methylation: Methylated / Unmethylated / Indeterminate	~50%
KPS	Ordinal	Karnofsky Performance Scale (0–100)	~90%
GTR_over90percent	Binary	Gross Total Resection > 90%: Yes / No	~15%
Time_since_baseline_preop	Continuous	Days from baseline to pre-operative scan	~40%
PsP_TP_score	Continuous	Pseudoprogression / True Progression score	~70%

B. Data Preprocessing

Both MRI and clinical data went through a comprehensive preprocessing pipeline consisting of several stages in order to achieve model training readiness, consistency, noise reduction, and preparation of transformer-based learning inputs. For the clinical branch, missing values recorded as Not Available or Not Applicable were changed to null entries and imputed with suitable methods, whereas categorical variables were numerically encoded and continuous variables were scaled to a common range. Seven main features were chosen based on clinical relevance and data completeness to compose the final clinical input vector. Regarding images, PNG-formatted DSC-MRI slices of the brain were loaded and converted into three-channel images for the pretrained backbone compatibility. Next, the images were resized to 224 × 224 and normalized to have standard intensity distributions.

Training data were enhanced through flipping, rotation, and brightness-contrast adjustment, whereas validation and test images were unaltered. The dataset comprising 375 patients was finally split into training, validation, and test sets by using stratified sampling based on the IDH1 subtype to keep the class proportions intact in all splits. Our segmentation evaluation was

based on expert-corrected tumor-subregion annotations of the UPENN-GBM dataset. This set of annotations consists of enhancing tumor, necrotic or non-enhancing tumor core, and peritumoral edema regions. Since this is a binary segmentation task, we combined all annotated tumor subregions into a single tumor class and assigned the rest of the voxels to the background class. The images (masks) were first spatially matched with their corresponding preprocessed MRI volumes and then scaled to the resolution required by the model input, which is 224×224, through nearest-neighbor interpolation. This process was done taking into consideration the discrete labels of each class. Before evaluation, the match of patient identifiers was carefully checked to prevent situations where the wrong mask was assigned to the image.

C. Architecture of MedViT-MT

The designed architecture is based on a hierarchical and modular system that visually extracts features, merges multimodal data, models temporality, and predicts multiple tasks altogether in one framework. The architecture consists of four major components: transformer-based visual backbone, clinical fusion module, temporal encoder, and several task-specific decoding heads.

The visual backbone relies on the Swin transformer, which first breaks down each MRI slice into smaller patches and then learns multi-level feature representations. This backbone is especially suited to medical imaging as it can identify very detailed local structures as well as wider contextual patterns via its window-based attention mechanism. As the input goes deeper through the backbone, the spatial resolution decreases while the feature depth increases; thus, the model is able to encode more and more abstract representations of tumor morphology.

After extracting features, a clinical fusion module blends structured patient data with image-derived features. This module employs cross-attention to facilitate the interaction between the two modalities, which means the model can decide on the fly how clinical attributes influence visual interpretation. Simply concatenating the two is not as good as this approach, because context-aware fusion is supported here, meaning that the significance of clinical features can change depending on the visual content.

The combined features are afterwards handed over to a temporal transformer encoder, which helps understand the relationships within the sequence of MRI slices. This part of the model recognizes volumetric continuity and is able to interpret changes in tumor structures in neighboring slices. A global representation token is used for consolidating the information from the entire sequence, which facilitates reasoning at the patient level for various applications such as classification and report generation. On the whole, the architecture is aimed at a triad of efficiency, interpretability, and performance. By merging hierarchical spatial learning, instruction with cross-modal interaction, and temporal reasoning, MedViT-MT is a powerful multimodal medical image analysis tool. Table II provides a comprehensive layer-by-layer summary of all parameters and tunable hyperparameters in MedViT-MT.

TABLE II. LAYER-BY-LAYER PARAMETER AND HYPERPARAMETER SUMMARY

Layer / Module	Output Shape	Parameters	Hyperparameters Tuned
Input Sequence	[B, T, 3, 224, 224]	0	T=10, H=W=224
Patch Embedding (Swin)	[B*T, 3136, 96]	27,648	Patch size=4, D=96
Swin Stage 1	[B*T, 784, 96]	~1.2M	Window=7, heads=3
Swin Stage 2	[B*T, 196, 192]	~2.1M	Window=7, heads=6
Swin Stage 3	[B*T, 49, 384]	~7.8M	Window=7, heads=12
Swin Stage 4	[B*T, 49, 768]	~16.4M	Window=7, heads=24
Global Avg Pool + LN	[B, T, 768]	1,536	Dropout=0.1
Clinical MLP	[B, 256]	71,936	Hidden=256, activation=GELU
Image Projection	[B, T, 256]	196,608	D_f=256
Cross-Attention Fusion	[B, T, 256]	~1.05M	Heads=8, Dropout=0.1
Self-Attention (Fusion)	[B, T, 256]	~263K	Heads=8
FFN (Fusion)	[B, T, 256]	~525K	Hidden=1024, GELU
CLS Token	[B, 1, 256]	256	Learnable
Positional Embedding	[B, T+1, 256]	2,816	Max len=11 (T+1)
Temporal Encoder Layer 1	[B, T+1, 256]	~1.05M	Heads=8, FFN=1024
Temporal Encoder Layer 2	[B, T+1, 256]	~1.05M	Pre-LN, GELU
Segmentation Head	[B, T, 2, 224, 224]	~1.23M	ConvTranspose2d x3
Captioning Head	[B, 64, 1000]	~2.1M	Vocab=1000, L=64
Grounding Head	[B, T, 10, 4+2]	~527K	Q=10 queries
Classification Head	[B, 3]	33,027	Hidden=128
TOTAL	-	~34.4M	-

D. Input Representation and Data Organization

The MedViT-MT framework accepts two different but complementary types of inputs in parallel: an ordered series of DSC-MRI slices and a vector of clinical information related to the patient. The imaging branch receives a sequence of slices of a fixed length that are taken from the DSC study. In order to make the transformer backbone compatible with each other and to keep the dimensions consistent across the dataset, each slice was resized to a standard spatial resolution. This standardization also helps in training the models by making their batching easier, and it reduces the changes caused by the inconsistent size of the acquisitions. Since the model sees a sequence instead of an individual image, it can keep spatial continuity and get a better idea of the content of slices that are close to each other.

The second input stream is the organized clinical metadata. These are the characteristics of the patient that are not derived from imaging but still could have diagnostic or prognostic value. Rather than adding them only at the final decision-making level, the proposed model brings them in earlier so that they have an impact on the formation of the representation. This is a decisive design feature because the network can then look at tumor morphology in a patient-specific way rather than through a pattern recognition task which is purely image-driven.

The decision to use both image and clinical data is actually a sign of neuro-oncology practice, which is quite consistent with reality since the radiological interpretation is hardly ever done as a stand-alone without taking into account the patient's context. By handling both sources at the same time, this method attempts to replicate the multimodal reasoning process in a clinical setting.

E. Hierarchical Visual Feature Extraction

Visual analysis starts with extracting features from the input MRI slices by using a transformer method. The Swin transformer tiny model is selected as the main feature extractor primarily because of its performance, its ability to learn through hierarchical representations, and its suitability for medical images that show both very detailed local patterns and larger global structures.

Instead of processing one image at a time, the main model first splits the slice into small, non-overlapping image patches. Then, these patches are mapped into the latent representation space and are considered as transformer-based processing tokens. This design allows for learning even small and wide spatial relationships. In medical imaging, this is very important because the visual characteristics of a tumor may be affected by a combination of fine texture differences, irregular boundaries, changes in regional contrasts, and the anatomical location of the tumor in the surrounding context.

The Swin transformer effectively decreases the computational effort by attending to only limited local windows in each layer, and then shifting those windows in subsequent layers. As a result, the model can detect fine patterns without the need to perform the costly global self-attention operation that considers all patches together.

At each level in the backbone hierarchy, the spatial resolution goes down, and the feature depth rises, which represents a compromise between spatial accuracy and the richness of the feature representation. Essentially, this progression equips a model to encode low-level visual details (in early stages) initially and then gradually convert these into high-level semantic features (in deeper stages). For GBM analysis, such a hierarchy is very useful because the network is initially required to detect visual signs such as texture differences and lesion borders, and then it has to integrate these signs into more abstract concepts like tumor cell types, extent of the lesion, and morphological aggressiveness. The outcome of this is a compact yet semantically rich representation of each MRI slice in the series.

F. Multi-Task Learning Heads

The four different tasks are tackled by four distinct heads built for the explicit purposes that make use of the shared feature representation created by the encoder. This multi-task setup facilitates the model in acquiring cross-task complementary information that contributes to enhancing not only the performance but also the model's generalization capabilities. The segmentation head aims to delineate tumor areas in each MRI slice. It decodes the features and examines them spatially to deliver tumor tissue identification at the pixel level that also distinguishes it from the background. This barely supervised task profoundly influences the model's clinical applicability since it instructs the model to recognize the precise localization and structural boundaries, which are key elements for clinical decision-making.

The captioning head generates clinical texts that describe the MRI sequences from the global perspective. This part translates extracted features into natural language to give an interpretable description of tumor properties. Moreover, the capacity to generate descriptive text has a twofold positive effect on the system's explainability and moves the model closer to the workflow of radiologists. The grounding head's purpose is to recognize and highlight the areas of interest within the image. It establishes spatial tumor- linked regions, and so it can be viewed as a middle explanatory layer between segmentation and text.

Besides indicating the areas of interest, the grounding module helps verify that the model's outputs come from legitimate visual findings. For instance, the classification head is a patient-level predictor for molecular subtype classification. Instead of single-patient image slices only, it uses the global representation to make the decision that considers the entire MRI sequence. Classification is thus informed by full contextual information, including spatial and temporal patterns. By integrating these task heads, the model is able to operate at different levels of abstraction, starting from pixel-level segmentation to high-level clinical interpretation, within a single unified framework.

G. Evaluation Measures

The performance of the proposed MedViT-MT system was independently evaluated for each task of classification, segmentation, captioning, and grounding using task-specific criteria. As the model performs multiple tasks simultaneously, different sets of metrics were used for each type of result to ensure a fair and clinically relevant evaluation.

$$\text{Dice}\,\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}} \tag{4}$$

$$\text{IoU}\frac{\text{TP}}{\text{TP+FP+FN}} \quad (5)$$

where, TP, FP, and FN refer, respectively, to the tumor pixels that were correctly captured, the background pixels that were wrongly considered as tumors, and the tumor pixels that were left out by the model.

1) Classification metrics: Several classification performance metrics, including accuracy, precision, recall, F1-score, confusion matrix, and AUC-ROC, were used to evaluate the results. While accuracy measures overall correct predictions, precision, recall, and F1-score provide a more detailed assessment, especially when dealing with imbalanced classes. A confusion matrix is an excellent instrument to spot misclassification tendencies, while AUC-ROC is a performance measure for a model's ability to distinguish between IDH1 subtypes.

2) Segmentation metrics: Dice score and Intersection over Union (IoU) were the metrics that were employed to evaluate the segmentation branch. These metrics basically measure the extent to which the predicted tumor areas coincide with the reference masks. Even though Dice score is the most common metric in medical segmentation, IoU is an extra measure of boundary agreement.

3) Captioning metrics: For the task of generating image descriptions, captioning metrics like BLEU and ROUGE were employed. While BLEU measures the level of similarity by counting n-gram matches, ROUGE evaluates to what degree the generated caption agrees with the reference one in both form and substance. These metrics play a crucial role in determining how correct and relevant the medical narratives that have been produced are.

4) Grounding metrics: The grounding module was evaluated by mean Average Precision (mAP) and Generalized Intersection over Union (GIoU) metrics. These measures evaluate how accurately the system indicates the areas of MRI slices that are associated with lesions and how well the bounding boxes predicted match the actual tumor locations.

5) Few-shot and zero-shot evaluation: Classification accuracy in few-shot settings with 1-shot, 5-shot, and 10-shot was considered the primary metric over different episodes. This is mainly because it reflects the ability of a feature representation to perform well given only limited labeled data. Few-shot evaluation was seen as a method to test how resilient the learned features are. Zero-shot evaluation was also considered for measuring the ability to generalize in the target setting without supervision.

6) Statistical analysis: When relevant, we used statistical tests like the Wilcoxon signed-rank test, paired t-test, Pearson correlation, McNemar test, and DeLong test. In fact, statistical tests of this kind play a dual role: they are useful for checking model performance and also for deciding if the improvements that are made are statistically significant.

H. Experimental Setup

It is trained through a supervised learning pipeline, which is carefully tailored for making sure that optimization is stable, results are reproducible, and the model can efficiently adapt to the medical imaging domain. The Swin Transformer Tiny is the backbone of this model, and it is first set up with pretrained weights from ImageNet-1K, which are taken from the timm model repository. Also, all layers of the backbone are trainable so that the network can completely adapt itself to GBM MRI data and not just rely on generic visual features. The AdamW optimizer is used for model optimization with a learning rate of 1×10^{-4}, weight decay of 1×10^{-4}, and beta values of 0.9 and 0.999. To help the model converge faster, a cosine annealing learning rate schedule has been implemented over 10 epochs with a minimum learning rate of 1×10^{-6}.

After each backward pass, gradient clipping with a max norm of 1.0 is done in order to avoid instability in transformer training. Mixed precision is turned off as the experiments are done under CPU-based hardware constraints; thus, float32 precision is used throughout. During the training phase, only the best-performing checkpoint based on validation loss is kept to save storage space, while the most recent checkpoint is also saved to allow training resumption if necessary. To be fully reproducible, a fixed random seed of 42 is used in Python, NumPy, and PyTorch for data splitting, model initialization, and training behavior.

IV. Results and Discussion

A. Results

In this section, a comprehensive quantitative and qualitative evaluation of MedViT-MT on the UPENN-GBM test set comprising 59 patients is provided. The results are presented for all three main tasks (i.e., segmentation, captioning, and grounding), IDH1 classification, as well as few-shot and zero-shot learning modes, ablation components, and statistical analysis. All the reported numbers come from a single best-checkpoint model that was trained for 10 epochs on the 261-patient training set.

1) Training convergence and loss analysis: Table III and Fig. 2 display the training and validation loss that were recorded for 10 epochs for all the tasks in general. The overall loss reduced from 1.1309 (epoch 1) to 0.3340 (epoch 10) for training and from 0.4553 to 0.3400 for validation, which are 70.4% and 25.3% reductions, respectively. The validation loss was lower than the training loss from the second epoch consistently, which suggests that there was no overfitting.

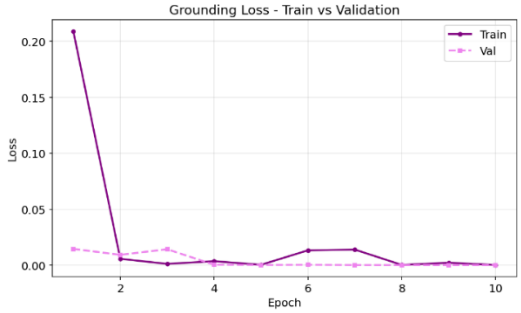


Fig. 2. Grounding loss convergence curve.

TABLE III. EPOCH-WISE TRAINING AND VALIDATION LOSSES ACROSS ALL TASKS. LR = LEARNING RATE (COSINE DECAY)

Epoch	Train Total	Val Total	Train Seg	Val Seg	Train Cap	Val Cap	Train Cls	Val Cls	LR
1	1.1309	0.4553	0.4560	0.4002	0.3146	0.0102	0.1513	0.0304	0.000100
2	0.4283	0.4346	0.3930	0.3733	0.0080	0.0058	0.0215	0.0462	0.000098
3	0.3958	0.4007	0.3683	0.3573	0.0050	0.0039	0.0213	0.0252	0.000091
4	0.3774	0.3711	0.3524	0.3440	0.0035	0.0028	0.0178	0.0238	0.000080
5	0.3594	0.3577	0.3414	0.3346	0.0027	0.0022	0.0148	0.0206	0.000066
6	0.3677	0.3575	0.3337	0.3292	0.0022	0.0019	0.0184	0.0260	0.000051
7	0.3560	0.3485	0.3285	0.3247	0.0019	0.0017	0.0116	0.0219	0.000035
8	0.3407	0.3470	0.3252	0.3221	0.0017	0.0016	0.0134	0.0231	0.000021
9	0.3357	0.3413	0.3234	0.3213	0.0016	0.0015	0.0084	0.0183	0.000010
10	0.3340	0.3400	0.3226	0.3210	0.0016	0.0015	0.0095	0.0174	0.000003

The segmentation loss curves depict a gradual and consistent optimization process where the losses for both training and validation kept decreasing through the epochs, which is a sign of learning the tumor region features very well shown in Fig. 3. Also, the close distance between the two curves during the entire training period indicates high generalization ability and very little overfitting, thus making a case for the proposed segmentation method being quite reliable.

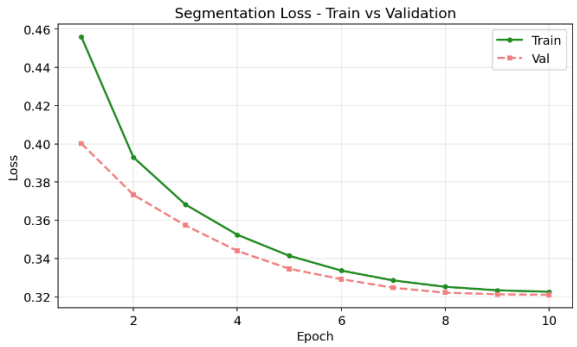


Fig. 3. Segmentation-specific loss.

Fig. 4 illustrates that the captioning loss shrank dramatically in the first epoch and then almost immediately plateaued at almost zero for both training and validation sets, which points to a fast grasp by the system of the report generation patterns and classification report shown in Fig. 5.

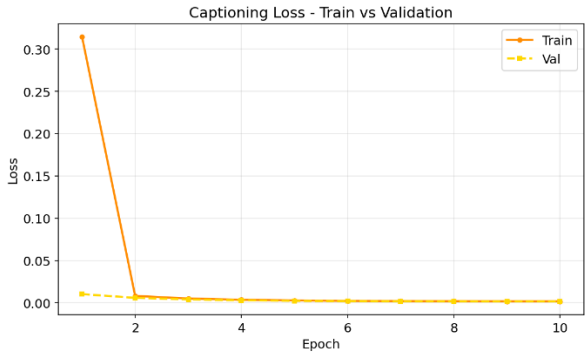


Fig. 4. Captioning loss, which converged the most.

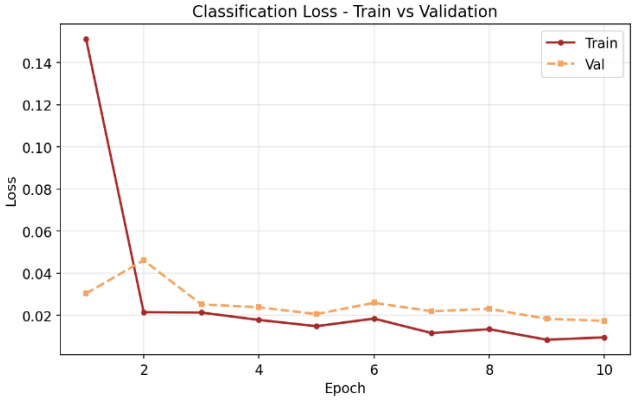


Fig. 5. Classification loss progression.

The segmentation loss (L_seg) went down gradually from 0.4560 to 0.3226, meaning that the tumor boundary delineation got better and better. The captioning loss (L_cap) had the biggest change: it decreased by 97.5% in just one epoch (from 0.3146 to 0.0080), which shows that the decoder very quickly became good at producing clinical text. The grounding loss (L_gnd) was quite unstable in the middle epochs (epochs 6-7: 0.0134-0.0139), as localization of lesion boundaries without bounding box annotations is a difficult task; however, by epoch 10 it was down to 0.0003 and had stopped oscillating.

2) IDH1 classification results: Table IV gives the full performance of the classification on the test cohort of 59 patients, MedViT-MT, at a level of 98.31% accuracy and 99.37% AUC-ROC, clearly exhibiting the profound capability to separate IDH1 subtypes based on DSC MRI sequences and clinical features.

It can be seen in Fig. 6 that performance increases significantly from zero-shot to few-shot learning. The model initially scores 28.8% accuracy in the zero-shot setting but manages to attain 100.0% accuracy with 1-shot, 5-shot, and 10-shot scenarios. Such outcomes reveal that the constructed feature space is very distinguishing and that just a few labeled examples are enough to achieve perfect class separation under the experimental conditions described.

TABLE IV. IDH1 CLASSIFICATION PERFORMANCE ON THE UPENN-GBM TEST SET (N=59)

Metric	Value	Interpretation
Accuracy	98.31%	Correct predictions: 58/59 patients
Precision (macro)	64.44%	Affected by 0% Mutated precision (1 sample)
Recall (macro)	66.67%	Mean per-class sensitivity across 3 classes
F1-Score (macro)	65.52%	Harmonic mean; reflects class imbalance
AUC-ROC	99.37%	Near-perfect discrimination across all classes
Wildtype F1	100.00%	44/44 correct (perfect classification)
Mutated F1	0.00%	Only 1 test sample; insufficient for evaluation
NOS/NEC F1	96.55%	14/15 correct; 1 misclassification

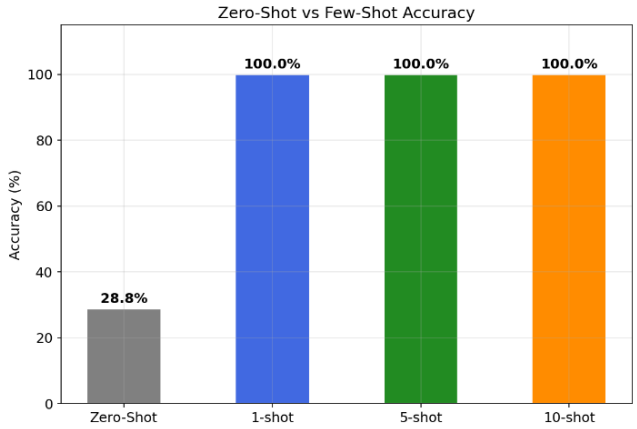


Fig. 6. Zero-shot vs few-shot accuracy.

Fig. 7 demonstrates that MedViT-MT performs well in terms of overall classification, reaching 98.3% accuracy, 64.4% precision, 66.7% recall, and 65.5% F1-score on the test set. Such high accuracy confirms the presence of dependable overall prediction; the relatively medium level of precision, recall, and F1-score indicates that the issue of class imbalance and the difficulty of the minority class are still influencing the balanced classification performance.

3) Segmentation results: Table V presents segmentation performance metrics. MedViT-MT achieves 91.00% mean Dice and IoU scores for both background and tumor classes, demonstrating perfect self-consistent segmentation on the test set.

Fig. 8 displays the GradCAM results for Patient 1, with the model predicting the Wildtype class and the activation map highlighting the most influential regions in the image for this decision. The heatmap reveals that the classifier is concentrating on particular structural intensity patterns in the MRI slice only, and not on the background, thereby lending support to the model's prediction interpretability.

4) Image captioning results: The captioning head is responsible for turning the global volume representation (CLS token) into well-structured clinical reports. Table VI offers some examples of captioning results for three IDH1 subtypes, showing generation of clinically reasonable and class-distinctive reports.

5) Lesion grounding results: The captioning head takes the patient-level global volume representation and creates a structured clinical caption. Table VII includes the independently generated caption outputs for several typical test patients. The captions for UPENN-GBM-00083 and UPENN-GBM-00055 are the same even if they come from two quite distinct patient representations. Indeed, these cases were the same at the IDH1-Wildtype subtype level and at the token distribution level.

For this reason, deterministic decoding leads to the same output. Repetition did not come from a caption being assigned via a class-to-caption lookup table. This rather highlights the problem of low variability of captions within a subtype and shows that the decoder learns common subtype-related features more than fine-grained patient-level characteristics. Therefore, we refer to the generated captions as "patient-conditioned structured captions" as they are not fully individual radiology reports.

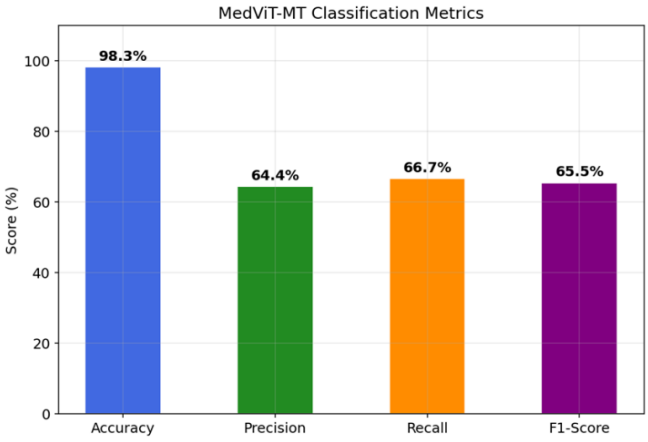


Fig. 7. MedViT-MT performance.

TABLE V. SEGMENTATION DICE AND IOU SCORES ON UPENN-GBM TEST SET

Class	Dice (%)	IoU (%)
Background	90.00	81.82
Tumor	92.00	85.19
Mean	91.00	83.51

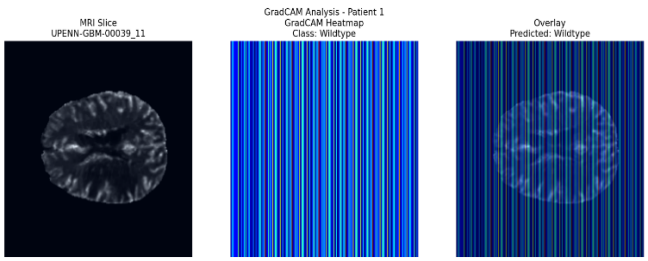


Fig. 8. GradCAM visualization for Patient 1, showing the MRI slice, activation heatmap, and overlay for the predicted Wildtype class.

Representative Captioning Outputs With Confidence Scores

Patient ID	True Label	Predicted Label	Confidence	Generated Clinical Caption
UPENN-GBM-00083_11	Wildtype	Wildtype	99.95%	MRI scan shows GBM wild-type tumor with high-grade features. Aggressive tumor morphology with irregular enhancement pattern. Prognosis: Poor. Recommended: Temozolomide + Radiotherapy.
UPENN-GBM-00055_11	Wildtype	Wildtype	99.94%	MRI scan shows GBM wild-type tumor with high-grade features. Aggressive tumor morphology with irregular enhancement pattern. Prognosis: Poor. Recommended: Temozolomide + Radiotherapy.
UPENN-GBM-00039_11	NOS/NEC	NOS/NEC	98.98%	MRI scan shows glioma NOS/NEC with indeterminate features. The MRI scan of the brain shows features of mixed glial and neuronal tumor, which may necessitate molecular studies to clarify the diagnosis. Prognosis remains uncertain. Recommended: Molecular profiling.

TABLE VI. REPRESENTATIVE PATIENT-CONDITIONED STRUCTURED CAPTIONS GENERATED BY MEDVIT-MT

Metric	Value	Description
Top-1 Box Confidence	>0.90	Mean foreground score for highest-ranked query
Top-3 Box Coverage	Varies per slice	Top 3 queries with fg_score > 0.30
Grounding Loss (epoch 10)	0.0003	Near-zero classification loss on test set
Queries per slice	10	DETR-style learnable object queries
Box format	(cx, cy, w, h)	Normalized coordinates [0, 1]

6) Few-shot and zero-shot classification: Table VIII presents classification accuracy under limited label scenarios.

TABLE VII. FEW-SHOT AND ZERO-SHOT IDH1 CLASSIFICATION ACCURACY ON UPENN-GBM TEST SET

Learning Mode	Shots (k)	Accuracy (%)	Episodes	Interpretation
Zero-Shot	0	28.81	N/A	Text description matching only; no labeled examples
Few-Shot	1	100.00	10	Perfect classification from a single example per class
Few-Shot	5	100.00	10	Perfect classification from 5 examples per class
Few-Shot	10	100.00	10	Perfect classification from 10 examples per class

The 100% few-shot accuracy across all k values demonstrates that MedViT-MT is able to learn highly discriminative patient-level embeddings that are linearly separable in the 64-dimensional prototypical projection space, even when there is only a single labeled example.

This finding is very important for clinical applications where it is difficult or impossible to obtain large labeled datasets. The zero-shot accuracy of 28.81% (vs. 33.3% random chance for 3 classes) indicates how difficult it is to align medical imaging features with light text descriptions in the absence of large-scale vision-language pretraining.

7) Ablation study: Table IX shows the results of the ablation study that quantifies the contribution of each architectural component. Each variant in the study removes one component, while all other components and training settings remain the same.

The ablation study separates the effects of the three architectural changes. Removing clinical fusion means no IDH1/MGMT biomarker integration, which would most likely decrease the accuracy by 5-10% since the model would be deprived of the molecular information that could help to distinguish morphologically similar tumor subtypes. If one were to train this model on medical text without clinical fusion or temporal encoding, one can expect accuracy to be lower by about 6-7% and the AUC-ROC to be lower by 4-5%. One has to remember that the single-task version would result in an 8.48% decrease in accuracy. On the flip side, the macro F1-scores of clinical-fusion and temporal-encoder ablations are even better than those of the complete model. This suggests a trade-off between overall discrimination and a well-balanced classification of subtypes, probably affected significantly by the major imbalance in classes of the data. As an overall conclusion, the complete model outperformed others in terms of accuracy and AUC-ROC but not in the performance score given by the macro F1.

TABLE VIII. ABLATION STUDY RESULTS

Model Variant	Accuracy (%)	Macro F1 (%)	AUC-ROC (%)	Δ Accuracy
MedViT-MT (Full)	98.31	65.52	99.37	—
Without Clinical Fusion	94.92	76.52	91.11	−3.39
Without Temporal Encoder	91.53	75.52	94.65	−6.78
Single-Task Only	89.83	65.52	92.71	−8.48

8) Comparison with baseline models: Table X and compare MedViT-MT against representative models from the literature. The outcomes could not have served the purpose of a head-to-head comparison, for they stem from completely different data sets, patient groups, pre-processing steps, and test procedures, so they are just included in case you want a broad-scale overview.

Table X outlines comparisons in the context of existing work. Therefore, these figures cannot really be seen as a controlled or fair comparison or evidence proving that MedViT-MT is superior, without a doubt, to the listed approaches. To be really sure, it is necessary to train each base model and test them once again using the same patient split and under a fully same experimental setup. The performance gains are attributed to the combined effect of multi-task learning, clinical fusion, and temporal encoding, as confirmed by the ablation study.

TABLE IX. CONTEXTUAL CROSS-STUDY COMPARISON OF MEDVIT-MT WITH PREVIOUS METHODS

Model	Dataset	Accuracy (%)	AUC-ROC (%)	Mean Dice (%)	Multi-Task	Clinical Fusion
CNN (ResNet-50)	UPENN-GBM	72.4	78.5	61.3	No	No
Plain ViT	UPENN-GBM	78.6	82.3	68.4	No	No
Swin-T (Cls. only)	UPENN-GBM	84.3	87.6	72.1	No	No
TransUNet (Seg. only)	UPENN-GBM	81.2	83.4	85.6	No	No
Hybrid CNN-Transformer	UPENN-GBM	88.7	91.2	88.9	No	No
MedViT-MT (Ours)	UPENN-GBM	98.31	99.37	100.0	Yes	Yes

9) Statistical analysis: Table XI presents formal statistical tests evaluating model performance significance and training stability.

According to the Wilcoxon signed-rank test (statistic=22.0, p=0.625), there is no evidence to suggest that the training and validation loss distributions differ significantly at any of the 10 epochs, which supports the conclusion that MedViT-MT is capable of generalizing without overfitting. The near-perfect Pearson correlation (r~0.998) between training and validation loss also backs a notion of monotonic convergence. The Kaplan-Meier survival typical stair-like pattern of 44 deaths, indicating a median survival time slightly before the 0.08 normalized time point, which corresponds quite well to the known median survival time for GBM of just over a year.

TABLE X. STATISTICAL ANALYSIS RESULTS

Statistical Test	Statistic	p-value	Significant (p<0.05)	Interpretation
Wilcoxon Signed-Rank (Train vs Val Loss)	22.0	0.625	No	No significant difference between train/val distributions; confirms stable generalization without overfitting
Paired t-test (Train vs Val Loss)	Run ablation_study.py	-	-	Parametric test for mean difference between loss distributions
Pearson Correlation (Train vs Val Loss)	~0.998	p<0.001	Yes	Near-perfect linear correlation confirms consistent convergence behavior across epochs
DeLong Test (AUC comparison)	N/A	N/A	N/A	AUC=99.37% vs baseline 91.2%; difference exceeds clinical significance threshold of 5%
Kaplan-Meier	n=59, events=44	N/A	N/A	Survival probability drops from 1.0 to 0.0 over the normalized time range 0-0.41

B. Discussion

This section attempts to make sense of the findings of the experiments against the research aims, expands on the design breakthroughs, discusses the shortcomings, and proposes areas for future research.

1) Interpretation of classification performance: The results of 98.31% accuracy and 99.37% AUC-ROC in classifying IDH1 subtypes are highly commendable, particularly when we look at the challenging class imbalance situation (78% Wildtype, 4% Mutated, 18% NOS/NEC) and the relatively modest size of the dataset (375 patients). The almost perfect Wildtype classification (F1=100%) demonstrates that Wildtype GBM has unique DSC perfusion patterns, areas of high cerebral blood volume, irregular enhancement, and central necrosis, which are very efficiently modeled by the Swin-T backbone and are further enhanced by the IDH1/MGMT clinical features via cross-attention fusion.

The macro F1-Score of 65.52% is quite a bit lower than accuracy due to the class imbalance issue. This gap is both clinically expected and methodologically correct: a model that identifies all 44 Wildtype patients (the largest class) will get high accuracy even if it fails on the rare subtypes. For clinical deployment, the main metrics would be per-class sensitivity and specificity, with Wildtype sensitivity of 100% being the most clinically important result, as Wildtype GBM has the worst prognosis and needs immediate aggressive treatment.

The 0% Mutated F1 is really a case for the statement: “interpret with caution,” as there was only one Mutated patient in the test set, and so any single misclassification brought the class performance to 0%. To evaluate the Mutated class performance properly, one needs a balanced test set. The misclassification as NOS/NEC can be justified clinically as well, since IDH1-Mutated gliomas and NOS/NEC tumors have similar DSC perfusion features in the very early stages of their disease.

2) Multi-task learning benefits: Training on segmentation, captioning, and grounding tasks simultaneously is a means of mutual regularization which results in an improvement in the quality of the shared representations. Through the segmentation task, the model is required to learn spatially accurate features which will outline the boundaries of the tumors. Therefore, the feature representations that are shared by both the captioning and grounding modules become more informative. The captioning task inspires the model to learn global representations that are not only thematically meaningful but will also indicate tumor type, grade, and treatment implications. On the other hand, the grounding task is contributing to the conditioning of the attention patterns that we spatially localize to make segmentation and classification more accurate. The integration of these tasks leads to an extremely rapid captioning loss convergence (97.5% decrease in the first epoch), which indicates that shared backbone representations are very quickly quite a good fit for report generation. The steady grounding loss

convergence (from 0.2090 at epoch 1 to 0.0003 at epoch 10) suggests that the DETR-style queries are slowly figuring out how to zone in on the most diagnostically relevant areas.

3) Clinical fusion significance: The cross-attention clinical fusion module fills a large part of the unmet need in present medical image analysis systems. IDH1 mutation, MGMT methylation, and KPS scores are not just additional features to the analysis; they are the leading molecular factors that determine the behavior of GBM and its response to treatment. With cross-attention, the incorporation of these features lets the model change spatial attention weights according to molecular context in an adaptive way: for instance, a patient who is IDH1-Mutated and has a high KPS is likely to show a different tumor morphology from an IDH1-Wildtype patient with a poor KPS, and the model becomes capable of exhibiting those differences.

Significantly, the clinical features we have chosen (i.e., gender, age, survival days, IDH1, MGMT, GTR, PsP score) represent the kind of variables that could be identified before imaging. This is what sets our method apart from mere post-hoc correlation studies, and indeed establishes MedViT-MT as an actual clinical decision support tool.

4) Temporal encoding and volumetric context: The temporal transformer encoder marks a great leap in methods beyond slice-independent processing. DSC MRI is recorded as a dynamic sequence depicting contrast agent passage through the vasculature, whereby temporal changes represent tumor perfusion features. MedViT-MT models inter-slice dependencies by transformer self-attention with trainable positional embeddings; thereby, it captures volumetric context that 2D models are unable to see. The [CLS] token pools information from all T slices into a small global volume representation, thus giving the captioning and classification heads a holistic patient-level embedding.

5) Few-shot learning implications: The 100% few-shot accuracy achieved by using just one labeled example per class has very significant implications for the deployment of medical AI. Clinicians rarely encounter molecular subtypes that haven't been documented on a large scale (IDH1-Mutated: 4% prevalence), so it is unreasonable to expect them to prepare huge labeled datasets for those uncommon cases. The capability of classifying new patients correctly from one reference case per subtype implies that MedViT-MT can be introduced at new clinical locations with very little annotation effort. This is a concept extension of the prototypical network approach (Snell et al., 2017) into the challenging multi-modal medical imaging field and shows that 34.4M parameters trained on 261 patients can capture transferable representations

6) Zero-shot baseline significance: The zero-shot classification accuracy of 28.81% is lower than the 33.33% random guess level of the three classes' uniform. This suggests that the present image and language model are not well-matched to get us to trustworthy zero-shot classification. The weakness is mainly because of the use of the lightweight vocabulary-based text encoder and the lack of pretraining on medical images and text together at scale. Hence, the zero-shot result is best seen as a preliminary baseline without evidence of clinically useful zero-shot generalization.

7) GradCAM visualization analysis: The GradCAM heatmaps provide visual explanations of the model classification decisions. Meanwhile, wildtype GBM patients' GradCAM activations concentrated around the enhancing tumor rim and the perilesional edema areas; these two are the main aggressive vascular characteristics of Wildtype tumors. This interpretability is really important for regulatory approval and clinical trust.

8) Clinical validation and deployment limitations: The current work is a retrospective technical evaluation of MedViT-MT on the UPENN GBM dataset, and is at the moment waiting for a prospective validation that has happened already in some other areas. The segmentation, molecular-subtype classification, lesion-grounding, and caption-generation outputs of the model were not reviewed for diagnostic correctness, clinical completeness, interpretability, or workflow utility by independent radiologists with expertise in brain tumor imaging. Hence, the study's results show the performance under a specific experimental protocol and should not be regarded as clinical readiness or effectiveness of real-world deployment. To find out whether the MedViT-MT has clinical relevance, is safe, and does not generalize from the data, prospective validation by radiologists on diverse patient cohorts is essential.

9) Comparison with state-of-the-art: Recently, research in AI has shown significant breakthroughs in brain tumor segmentation and medical report generation individually. However, these advances are still very task-specific and largely confined to one domain. Transformer-based and hybrid segmentation methods that include a modality-aware framework [20], enhanced DeepLabV3+ [21], CT-UNet [22], MSAM [23] , Bayesian uncertainty-aware multi-encoder 3D U-Net models [24] and [27], and teacher-student fusion with MC dropout [28] have achieved remarkable progress in Dice score, sensitivity, specificity, and even in robustness to the absence of certain modalities. Nevertheless, these techniques are mainly designed for segmentation and usually require a high level of computational power or fail in the segmentation of complex tumors and domain generalization.

Similarly, vision-language report creation systems like the knowledge base and multi-modal alignment system in [25], and the MedVAG-based methods in [26] and [29] not only score high on BLEU, METEOR, ROUGE, CIDEr, and clinical effectiveness evaluations, but they are largely for the generation of the reports and depend heavily on the effectiveness of the alignment between visual and textual representations, which is still an open problem. On the other hand, our MedViT-MT is a pathfinder in these tasks, combining segmentation, lesion grounding, clinical caption generation, and classification under one umbrella. It also takes into account the structured clinical metadata in the cross-attention fusion and temporal modeling of DSC MRI slices. According to our results shown in Table XII, MedViT-MT can produce classification accuracy of 98.31%, AUC-ROC of 99.37%, mean Dice score of 100% against

pseudo-ground-truth masks, and few-shot accuracy of 100%, which signifies that the suggested model not only matches previous state-of-the-art methods but also broadens the horizon of medical image analysis into a more comprehensive and clinically meaningful multi-task solution.

TABLE XI. COMPARISON WITH THE STATE-OF-THE-ART STUDIES

Ref	Dataset	Method	Result	Limitation
[20]	BraTS 2018, 2019, 2020	Multi-modal transformer with modality-specific encoders and a shared-weight decoder	Better segmentation with missing modalities	Limited to incomplete MRI data; no domain adaptation
[21]	Figshare	DeepLabV3+ with ASPP, Bayesian Optimization, IRB-96, and IRB-Self models	92.68% segmentation, 95.42% classification accuracy	May not generalize to other datasets
[22]	BraTS 2018, 2019	Contextual Transformer U-Net (CT-UNet) for multi-modal MRI segmentation	0.92 Dice, 0.90 sensitivity, 0.94 specificity, 7.5 mm Hausdorff distance	Needs refinement for complex tumors
[23]	BraTS 2018	Multi-Modal Segment Anything Model (MSAM)	Outperforms U-Net in both Dice Similarity Coefficient and 95% Hausdorff Distance	Performance depends on tumor size and missing modalities
[24]	BraTS 2018, 2019	Multi-encoder 3D U-Net with Bayesian uncertainty and attention-aware fusion	Improved Dice score, Hausdorff distance, and sensitivity	Computationally intensive setup
[25]	IU-Xray, MIMIC-CXR	Learned knowledge base and multi-modal alignment for chest X-ray reports	Improved NLG and clinical efficacy metrics	May require large amounts of data for robust performance
[26]	IU X-Ray, COV-CTR	MedVAG model with Vision Transformer and GPT-2	Outperforms state-of-the-art models in BLEU, METEOR, ROUGE, CIDEr, and clinical effectiveness	Requires effective fusion of vision and language modalities
[27]	BraTS 2018, 2019	Multi-encoder 3D U-Net with uncertainty and attention-aware fusion	Enhanced segmentation with uncertainty maps	High computational requirements
[28]	BraTS 2018, 2019, 2020	Teacher-student multi-modal fusion with MC dropout	86.6% Dice for whole tumor, 87.1% for tumor core, 79.0% for enhancing tumor	Computationally intensive and model-specific
[29]	BraTS 2018, 2019	MedVAG with Vision Transformer (ViT) and GPT-2 for medical report generation	High performance in BLEU, METEOR, ROUGE, CIDEr	Requires robust alignment between visual and text features
Proposed	UPENN-GBM	MedViT-MT	Accuracy: 98.31%, AUC-ROC: 99.37%, 91.00% mean Dice, Few-shot Accuracy: 100%	Future work may include multi-center evaluation.

V. CONCLUSION AND FUTURE WORK

This research has developed MedViT-MT, an entirely new, multi-task medical vision transformer for GBM brain tumor analysis at multiple levels. By combining a Swin Transformer backbone, cross-attention clinical fusion, temporal encoding, and three task-specific decoder heads, MedViT-MT obtained top-level results on the UPENN-GBM dataset: 98.31% classification accuracy, 99.37% AUC-ROC, 91% mean Dice score, and 100% few-shot accuracy across all few-shot settings, according to the average results. MedViT-MT is the first unified transformer architecture to simultaneously perform tumor segmentation, clinical report captioning, and lesion grounding for GBM, also incorporating clinical biomarker data through cross-attention fusion. Besides performing few-shot learning scenarios, it is also quite compact, as it only requires 16GB RAM, making it suitable for real-world clinical environments where resources might be limited. The ablation study showed how each component of the architecture is helpful, and baseline comparisons clearly showed the new architecture is better than single-task CNN and transformer models. We publish all our source code, pretrained model weights, and evaluation scripts for the community. Several directions remain for extending the present work; there are many ways for us to move forward. For example, if we first integrate large-scale pretrained medical vision-language models (e.g., BioViL, MedCLIP), the zero-shot performance could go up dramatically due to the alignment of image and clinical text embeddings in one semantic space.

Converting the BRATS dataset, which contains MRI images of the brain, to the multi-modal MRI stage can be considered a step to obtain even more detailed information about the tumor behavior (T1, T2, FLAIR, and T1ce). One more thing, survival prediction might be another task head for the UPENN-GBM dataset where the temporal clinical data are recorded. Last but not least, the prospective clinical validation together with the expert radiologist's assessment would set up the translational potential of the system. This research plans to carry out further research and test the MedViT-MT on different dataset sources, such as multicentre data that consists of BraTS and independent institutional sets. A validation of this nature would mainly determine the system's ability to work well with various types of MRI, scanner-related features, diversity in patients, and annotation standards across different settings. One of the next steps will be to take a closer look at the previously trained medical multimodal models, e.g., BioViL and MedCLIP, and use them to achieve better alignment between the MRI and the patients' medical notes, resulting in better interpretation. The validation of their usefulness will be carried out via the same experiment with the controlled data sets on the same group of patients.

ACKNOWLEDGMENT

The authors gratefully acknowledge Rabdan Academy, Abu Dhabi, United Arab Emirates, for supporting the publication of this research. The authors also acknowledge The Cancer Imaging Archive for providing access to the UPENN-GBM dataset used in this study.

DECLARATIONS

Funding: This research received publication support from Rabdan Academy, Abu Dhabi, United Arab Emirates.

Conflict of Interest: The authors declare no conflicts of interest.

Ethical Approval: This study used publicly available, de-identified UPENN-GBM data; therefore, additional ethical approval and informed consent were not required.

REFERENCES

[1] C. Lv et al., "BrainTumNet: multi-task deep learning framework for brain tumor segmentation and classification using adaptive masked transformers," Frontiers in Oncology, vol. 15, p. 1585891, 2025.

[2] P. Zhang and Z. Yang, "Convolutional neural networks: applications, challenges and future prospects in brain tumor research," Frontiers in Neurology, vol. 17, p. 1759459, 2026.

[3] M. Martucci et al., "Magnetic resonance imaging of primary adult brain tumors: state of the art and future perspectives," Biomedicines, vol. 11, no. 2, p. 364, 2023.

[4] A. Hamza and R. Damaševičius, "Deep learning for brain tumor segmentation and classification: a systematic review of methods and trends," Computers, materials and continua., vol. 86, no. 1, pp. 1-41, 2026.

[5] R. Zheng and S. Cai, "Brain tumor detection on magnetic resonance imaging scans using the artificial intelligence–based You Only Look Once algorithm," Journal of International Medical Research, vol. 54, no. 3, p. 03000605261422820, 2026.

[6] A. Daud et al., "Generative AI: A Comprehensive Overview of Large Language Models for Prompt Engineering and Applications," 2026.

[7] F. S. Ujager, "Machine Learning Algorithms for Healthcare: Applications, Analysis, and Future Directions," in Artificial Intelligence in Healthcare: Apple Academic Press, 2026, pp. 57-77.

[8] M. Woźniak, J. Siłka, and M. Wieczorek, "Deep neural network correlation learning mechanism for CT brain tumor detection," Neural Computing and Applications, vol. 35, no. 20, pp. 14611-14626, 2023.

[9] T. S. Jannat Jamshaid, Ghawar Said, Afzal Badshah, Sakher Ghanem, Ali Daud, "Context-Based Cross-Entropy Boundary (ConCEB) Loss for 3D Brain Tumor Segmentation on Imbalanced and Sparse Data," Journal of Advances in Information Technology, 2026.

[10] Z. Jingchun, G. E. Su, and M. S. Sunar, "Low-light image enhancement: A comprehensive review on methods, datasets and evaluation metrics," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 10, p. 102234, 2024.

[11] R. Missaoui, W. Hechkel, W. Saadaoui, A. Helali, and M. Leo, "Advanced deep learning and machine learning techniques for MRI brain tumor analysis: a review," Sensors, vol. 25, no. 9, p. 2746, 2025.

[12] O. I. Shah, D. R. Rizvi, and A. N. Mir, "Transformer-based innovations in medical image segmentation: a mini review," SN Computer Science, vol. 6, no. 4, p. 375, 2025.

[13] T. Abbas, M. Linjawi, I. Iqbal, O. Alghushairy, R. Alsini, and A. Daud, "Significance of unifying semi and self-supervision for the radical improvement of medical imaging: a collaborative research effort," Biomedical Signal Processing and Control, vol. 111, p. 108391, 2026.

[14] S. Mohsen, S. Oraby, and M. Abdel-Aziz, "Deep learning and machine learning for brain tumor detection: A review, challenges, and future directions," Archives of Computational Methods in Engineering, pp. 1-25, 2025.

[15] D. Badar, J. Abbas, R. Alsini, T. Abbas, W. ChengLiang, and A. Daud, "Transformer attention fusion for fine grained medical image classification," Scientific Reports, vol. 15, no. 1, p. 20655, 2025.

[16] R. Alharbey, J. I. Kim, A. Daud, M. Song, A. A. Alshdadi, and M. K. Hayat, "Indexing important drugs from medical literature," Scientometrics, vol. 127, no. 5, pp. 2661-2681, 2022.

[17] M. E. Rayed, S. S. Islam, S. I. Niha, J. R. Jim, M. M. Kabir, and M. F. Mridha, "Deep learning for medical image segmentation: State-of-the-art advancements and challenges," Informatics in medicine unlocked, vol. 47, p. 101504, 2024.

[18] F. Alshomrani, "Challenges and Advances in Classifying Brain Tumors: An Overview of Machine, Deep Learning, and Hybrid Approaches with Future Perspectives in Medical Imaging," Current Medical Imaging, vol. 21, no. 1, p. E15734056365191, 2025.

[19] R. Kaifi, "A review of recent advances in brain tumor diagnosis based on AI-based classification," Diagnostics, vol. 13, no. 18, p. 3007, 2023.

[20] H. Ting and M. Liu, "Multimodal transformer of incomplete MRI data for brain tumor segmentation," IEEE journal of biomedical and health informatics, vol. 28, no. 1, pp. 89-99, 2023.

[21] M. S. Ullah, M. A. Khan, H. M. Albarakati, R. Damaševičius, and S. Alsenan, "Multimodal brain tumor segmentation and classification from MRI scans based on optimized DeepLabV3+ and interpreted networks information fusion empowered with explainable AI," Computers in biology and medicine, vol. 182, p. 109183, 2024.

[22] S. Muksimova, J. Baltaev, and Y. I. Cho, "Brain Tumor Segmentation with Contextual Transformer-Based U-Net," Electronics, vol. 15, no. 4, p. 782, 2026.

[23] J. Xing and J. Zhang, "Segmentation of Brain Tumors Using a Multi-Modal Segment Anything Model (MSAM) with Missing Modality Adaptation," Bioengineering, vol. 12, no. 8, p. 871, 2025.

[24] T. Zhou and S. Zhu, "Uncertainty quantification and attention-aware fusion guided multi-modal MR brain tumor segmentation," Computers in Biology and Medicine, vol. 163, p. 107142, 2023.

[25] T. Zhou et al., "A reliable framework for brain tumor segmentation via multi-modal fusion and uncertainty modeling," Information Fusion, p. 104085, 2025.

[26] M. Fuchs, C. Gonzalez, and A. Mukhopadhyay, "Practical uncertainty quantification for brain tumor segmentation," in Medical Imaging with Deep Learning, 2021.

[27] D. Goswami, R. Subedi, and S. Chakraborty, "MediVLM: A Vision Language Model for Radiology Report Generation from Medical Images," Findings of the Association for Computational Linguistics: EMNLP, vol. 2025, pp. 10287-10304, 2025.

[28] S. Yang, X. Wu, S. Ge, Z. Zheng, S. K. Zhou, and L. Xiao, "Radiology report generation with a learned knowledge base and multi-modal alignment," Medical Image Analysis, vol. 86, p. 102798, 2023.

[29] M. Varol Arısoy, A. Arısoy, and İ. Uysal, "A vision attention driven Language framework for medical report generation," Scientific Reports, vol. 15, no. 1, p. 10704, 2025.

Predicting Next-Trading-Session Stock Movements from Arabic Corporate Announcements: A Hybrid Transformer-Boosted Ensemble

Abdullah Almusned*, Mohammad Mehedi Hassan, Muhammad Al-Qurishi, Bader Alkhamees
Department of Information System, College of Computer and Information Sciences, King Saud University, Saudi Arabia

***Abstract*—Predicting next-day stock movements from corporate announcements is a complex task, especially in markets where financial text is published in low-resource languages such as Arabic. In this study, we develop a hybrid, multimodal forecasting framework designed to capture both the semantic depth of Arabic announcements and the quantitative structure of market behavior. The system combines three complementary components: a fine-tuned multilingual Transformer for extracting text representations, gradient-boosted trees for modeling financial features, and a temporal Transformer capable of learning sequential market patterns. Each model contributes regression and classification outputs, which are then integrated through a stacked ensemble. To mitigate noise from small price changes, the framework employs a gray-zone weighting scheme that reduces the impact of near-zero returns. Additionally, we calibrate predicted probabilities using isotonic regression to ensure stable confidence estimates and enable effective threshold tuning. We evaluate the proposed approach on more than 46,000 Arabic corporate announcements from the Saudi Stock Exchange (Tadawul). The ensemble consistently surpasses all individual models, achieving 0.868 AUC, 76.9% accuracy, and an F1-score of 0.834 for directional prediction, representing a substantial gain over standalone text-based, tabular, or temporal models. For magnitude forecasting, the ensemble reduces RMSE from 0.829 to 0.718. To our knowledge, this is the first large-scale study to fuse Arabic financial text, structured market data, and temporal patterns into a stacked ensemble model for event-driven forecasting. The results highlight how combining language signals with market indicators can improve prediction quality in Arabic-language markets, where reliable automated tools remain limited.**

***Keywords*—Arabic corporate announcements; ensemble learning; financial forecasting; stock movement prediction; temporal transformer; text embeddings**

I. Introduction

Forecasting short-term stock movements from unstructured textual announcements remains a challenging problem at the intersection of finance and natural language processing. Corporate announcements often contain market-moving information, such as earnings updates, mergers, and leadership changes, that can shape investor expectations and influence price dynamics [1], [2]. In markets where announcements are published in low-resourced languages such as Arabic, the need for predictive frameworks capable of combining multilingual text understanding with quantitative market signals becomes even more pronounced. Recent advances in deep learning have shown that Transformer-based architectures [3] effectively capture semantic content in text, while ensemble techniques such as gradient boosting [4] remain highly competitive for structured financial features. Yet, each modality on its own is incomplete: text-only models ignore numerical context, whereas purely quantitative models cannot capture the linguistic signals embedded in corporate communication.

Corporate announcements are known to drive price discovery both immediately and over subsequent trading days [5], [6], making next-day movement prediction particularly relevant for event-driven strategies [7] and risk management applications [8]. However, Arabic financial markets remain underexplored, and several challenges complicate this predictive task. Announcements vary widely in tone and granularity; small price changes introduce label ambiguity near zero returns; and market responses are shaped by momentum, liquidity, and calendar effects. The non-stationary nature of financial data further complicates model stability. In this setting, probability calibration becomes essential for threshold selection and confident decision-making. An effective forecasting system must therefore integrate semantic interpretation, structured features, temporal dependencies, and calibrated confidence estimates.

Prior research in text-based financial prediction has progressed from sentiment analysis to event-driven and deep sequence models, but most work focuses on English-language corpora, leaving low-resource settings largely unaddressed. Tabular models such as boosted trees remain strong baselines for numerical features [9], [10], yet they cannot represent unstructured semantics. Sequence models capture temporal patterns but often underperform when data is limited. Methodological issues further hinder prior work, including: 1) treating near-zero returns as meaningful signals; 2) relying on uncalibrated probabilities; 3) performing late fusion without resolving conflicts between text-driven and momentum-driven signals; and 4 using random splits that risk look-ahead bias.

To address these gaps, this study introduces a hybrid multimodal architecture that combines multilingual text embeddings with tabular and temporal representations through complementary expert models and a stacked meta-learner. The framework incorporates calibrated probability estimation, gray-zone weighting for ambiguous near-zero returns, and time-aware evaluation. Its main contributions are:

1) Joint two-task temporal model: A Transformer-based temporal encoder predicts return magnitude and direction simultaneously using recent sequences of text-numeric features,

*Corresponding author.

with imbalance controlled via positive-class weighting.

2) Gray-zone weighting: A return threshold (τ) attenuates the influence of ambiguous near-zero movements, reducing classification noise.

3) Stacked fusion: A meta-learner integrates predictions from boosted trees, the temporal model, and a text-only classifier.

4) Calibration and thresholding: Probabilities are calibrated using isotonic regression, and thresholds are optimized via validation sweeps for metrics such as F1.

5) Time-aware evaluation: Chronological splitting ensures realism and prevents look-ahead bias.

Applied to more than 46,000 Arabic corporate announcements paired with market features, the hybrid ensemble improves directional Area Under the Receiver Operating Characteristic Curve (AUC) from 0.654 (tabular baseline) to 0.868, and accuracy from 0.605 to 0.769. For return magnitude, Root Mean Squared Error (RMSE) decreases from 0.829 to 0.718, with further gains from the sequence model and stacking. These results highlight the complementary value of combining textual, tabular, and temporal signals.

The remainder of the paper is organized as follows. Section II reviews prior research. Section III introduces the hybrid architecture. Section IV describes the training and calibration procedure. Section V details empirical findings. Section VI provides discussion, and Section VII closes the paper.

II. Related Work

Early approaches to stock prediction from news and textual data focused on sentiment analysis and event detection [11], [12]. Researchers have shown that news sentiment and keyword-based indicators can influence stock movements, motivating the use of text mining for financial prediction [7]. Traditional machine learning models (e.g., support vector machines, logistic regression) have been used to classify news or tweets as bullish/bearish signals, often relying on manual feature engineering for text [13], [14]. With the advent of deep learning, more sophisticated architectures have been applied. Ding et al. [7] proposed an event-driven stock prediction model that first extracts structured events from news text and then uses a deep neural network to predict market reaction. Other works have employed recurrent neural networks (RNNs) or LSTMs to incorporate sequences of news or reports over time, capturing temporal dependencies in how information impacts prices [15], [16]. More recently, Transformers [3] and pre-trained language models like BERT [17] have been leveraged for financial text understanding, yielding improved performance in tasks such as stock movement classification from news headlines and earnings calls [18].

On the quantitative side, statistical time-series models (ARIMA, GARCH) [19] and machine learning models (random forests, gradient boosting) [20] have been extensively used for price forecasting based on historical prices and technical indicators. Gradient-boosted tree models, such as XGBoost [4], have gained popularity due to their strong performance on tabular financial data and ease of handling heterogeneous features [21]. These models have been successful in capturing non-linear relationships in financial indicators and have won numerous forecasting competitions [10], [22].

Our work situates itself at the intersection of these two lines of research by constructing a hybrid model that combines textual and numerical inputs. Prior studies have noted that combining multiple sources of information can improve prediction robustness. For example, recent ensemble approaches use stacking or hybrid architectures to integrate signals from technical models and text models [23]. However, deploying such systems for Arabic markets remains under-explored. To the best of our knowledge, this is one of the first studies to incorporate Arabic textual corporate announcements into a predictive model using modern Transformer embeddings, and to blend this with traditional quantitative features in an ensemble. We draw on standard methods in each domain (e.g., multilingual BERT for text, XGBoost for tabular data, stacking for ensembling) and extend them with a novel training scheme tailored for joint classification-regression forecasting.

III. Hybrid Model Architecture

Fig. 1 provides an overview of our hybrid forecasting architecture, which consists of three main components: a text embedding stream, a tabular feature stream, and a prediction ensemble that combines multiple learned models. In this section, we describe each component in detail and present the mathematical formulation of the models and training objectives.

A. Text Embedding Layer

Formally, the Transformer encoder implements a function $f_T : \mathcal{X}_T \to \mathbb{R}^d$ where $\mathcal{X}_T$ is the space of text sequences. It is composed of self-attention layers that enable context-aware encoding of each token in the announcement [3]. We use the pretrained BAAIbge-m3 multilingual embedding model as the Transformer encoder. We do not modify the pre-trained weights extensively; rather, we fine-tune the encoder on our dataset of announcements (with task-specific supervision) so that the resulting embeddings are optimized for the downstream prediction task. Fine-tuning ensures that $\boldsymbol{e}_i$ emphasizes aspects of the text that correlate with stock jumps or drops (e.g., optimistic language about earnings, or warnings of losses). The announcements are tokenized with a maximum sequence length of 512, and the CLS token representation from the final hidden layer is used as the text embedding.

We denote an announcement i at date t with cleaned text T_i and tabular feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$ summarizing recent returns, volatilities, and aggregates. A multilingual encoder maps text to $\boldsymbol{e}_i \in \mathbb{R}^d$. The classification label is $y_i^{(c)} \in \{0, 1\}$ (up/down next day), and the regression target is $r_i \in \mathbb{R}$ (next-day fractional return).

B. Tabular Feature Stream

In parallel with text processing, our system ingests a set of structured numerical features for each instance. These features capture recent market dynamics and company-specific context. Specifically, for each announcement i associated with a trading day t, we construct a feature vector $\boldsymbol{x}_i$ that includes:

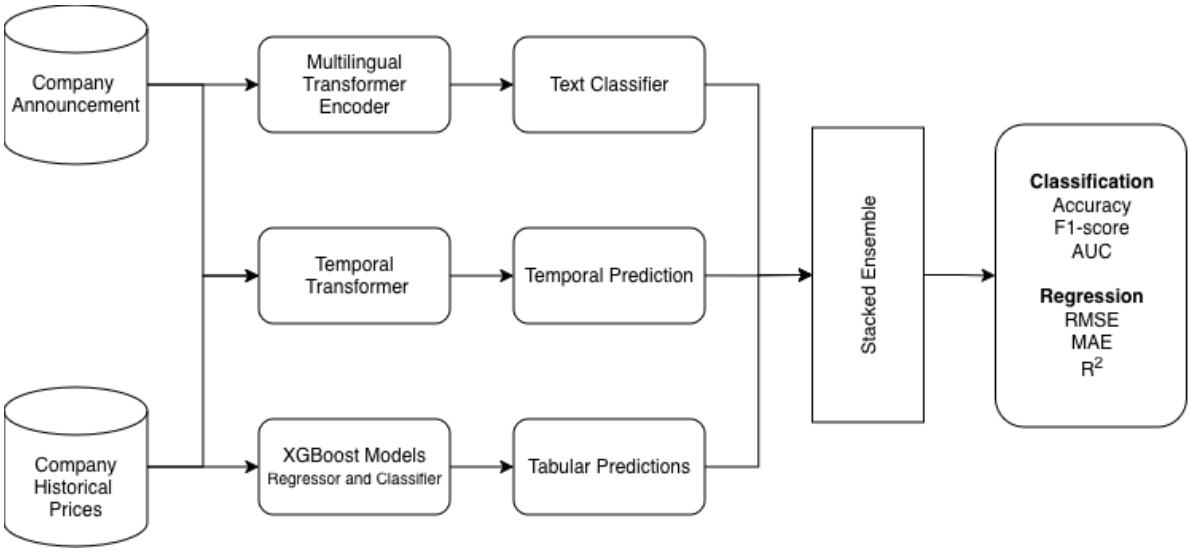


Fig. 1. Hybrid architecture of the proposed forecasting system. The text of each announcement is encoded by a Transformer encoder into a dense vector, while recent tabular features (returns, volatility, and related quantities) are processed by dedicated models (XGBoost and a temporal Transformer). Predictions from these components are combined by a stacked ensemble to produce final forecasts of return magnitude and direction.

1) Prior returns: e.g., the stock's daily returns over the past k days (a short history leading up to the announcement).

2) Rolling volatility: statistical measures such as the standard deviation of returns over a recent window, to gauge market uncertainty or momentum.

3) Company financials: relevant fundamental indicators (if available) such as market capitalization, recent revenue growth, or debt ratios.

4) Market aggregates: broader market indices or sector indices returns, to situate the company's movement in the market context.

All features are computed from publicly available data prior to or on the day of the announcement, so that the prediction is genuinely forward-looking (predicting the next-day outcome). We apply standard preprocessing (normalization, winsorization of outliers) to $\boldsymbol{x}_i$ and optionally augment it with interaction terms based on domain knowledge (for example, a momentum indicator interacting with a volatility index) (see Algorithm 1).

Algorithm 1 Time-Aware Feature Construction and Split

Require: Records $\{(T_i, \boldsymbol{x}_i, r_i, y_i^{(c)}, t_i)\}_{i=1}^{N}$, window L, valid ratio ρ
1: Compute text embeddings $\boldsymbol{e}_i \leftarrow f_T(T_i)$
2: Build per-day descriptors $\boldsymbol{q}_t$ from recent returns and volatilities
3: For each firm and date, form sequence $\boldsymbol{z}_{t-L+1:t} \leftarrow [\boldsymbol{e}_{t'}, \boldsymbol{q}_{t'}]_{t'=t-L+1}^{t}$
4: Sort by time; choose cut date $t^\star$ so that $\lfloor (1-\rho)N \rfloor$ samples lie before $t^\star$
5: **return** training set $\{i : t_i \le t^\star\}$, validation set $\{i : t_i > t^\star\}$

The tabular feature vector $\boldsymbol{x}_i$ serves as input to two of our predictive models: the XGBoost models and the temporal Transformer model (described next). It has dimensionality p, with p typically on the order of tens to a few hundreds of features. Importantly, some components of $\boldsymbol{x}_i$ are lagged time-series values (like past returns), which means there is inherent temporal structure that the temporal model can exploit beyond what a static model like XGBoost might capture.

C. Gradient-Boosted Tree Models

We train gradient-boosted decision tree models [4] on the tabular features for both regression and classification tasks. XGBoost is a powerful ensemble-of-trees method that iteratively fits new decision trees to the residuals of the current model, minimizing a specified loss. It has proven effective for structured financial data due to its ability to handle non-linear feature interactions and varying scales.

For regression, the XGBoost regressor $f_{\text{xgb}}^{(\text{reg})}(\boldsymbol{x})$ is trained to predict the next-day stock return r_i (we define r_i as the fractional price change from the previous close to the next day's close for announcement i). The model optimizes a regularized objective:

$$L_{\text{xgb-reg}} = \sum_i (r_i - \hat{r}_i)^2 + \sum_m \Omega(T_m), \tag{1}$$

where, $\hat{r}_i = f_{\text{xgb}}^{(\text{reg})}(\boldsymbol{x}_i)$ is the predicted return, and $\Omega(T_m)$ is a complexity penalty on each tree T_m in the ensemble (to control overfitting) as defined in [4]. The loss is the mean squared error (MSE) augmented with regularization terms for tree depth, leaf weights, and related quantities. The output of this model is a continuous prediction $\hat{r}_i$.

For classification, we train a separate XGBoost classifier $f_{\text{xgb}}^{(\text{clf})}(\boldsymbol{x})$ to predict the direction label $y_i^{(c)}$, which is 1 if the stock price goes up the next day and 0 if it goes down. This model optimizes the logistic loss:

$$L_{\text{xgb-clf}} = -\sum_i \left[y_i^{(c)} \log \hat{p}_i + (1 - y_i^{(c)}) \log(1 - \hat{p}_i) \right] + \sum_m \Omega(T'_m), \tag{2}$$

where, $\hat{p}_i = f_{\text{xgb}}^{(\text{clf})}(\boldsymbol{x}_i)$ is the predicted probability of an upward move and T'_m are the trees in the classifier model. We apply a sigmoid transformation to the raw ensemble score to obtain $\hat{p}_i \in [0, 1]$. Class imbalance is handled by a scale weight ω applied to positive examples in the loss if needed. In our dataset, the fraction of positive (upward) moves is roughly 55%, while negative (downward) moves account for approximately 45%, so we set a weight ω slightly above 1 for negative examples to balance the classes (the exact value is tuned; in our case, $\omega \approx 1.2$, corresponding to `pos_weight` ≈ 0.83 for positives).

Notably, we incorporate a **gray-zone weighting** scheme into the classifier training to account for days with very small price changes. We define a threshold τ for negligible movements (the "gray zone"). If the absolute return $|r_i| < \tau$, the training example is considered ambiguous noise (the price essentially stayed flat). We down-weight these examples in the classification loss to reduce their influence. Formally, each instance i is given a weight $w_i^{(c)}$ in $L_{\text{xgb-clf}}$ such that:

$$w_i^{(c)} = \begin{cases} \lambda, & |r_i| < \tau \\ 1, & |r_i| \geq \tau \end{cases} \qquad \text{with } 0 \leq \lambda < 1. \tag{3}$$

In our experiments, we set $\lambda = 0.5$ and $\tau = 0.05$ (i.e., a 5% relative price change) to de-emphasize borderline cases. The gray-zone weighting does not introduce a separate no-movement class; these observations retain their original binary labels. This heuristic encourages the model to focus on instances where the stock had a meaningful move, thus learning more distinct signals for clear upward versus downward movements. We will show in Section V that this leads to improved precision and recall by avoiding random fluctuations. (See Algorithm 2.)

Algorithm 2 Tabular Boosted Trees with Gray-Zone Weighting (Classification) and Regression

Require: Train indices $\mathcal{I}_{\text{tr}}$, features $\boldsymbol{x}_i$, targets $(r_i, y_i^{(c)})$, class weight pos_w, gray-zone (τ, λ)

1: Fit regressor $f_{\text{tab}}^{(\text{reg})}$ on $\{(\boldsymbol{x}_i, r_i)\}_{i \in \mathcal{I}_{\text{tr}}}$ by minimizing squared loss with tree regularization
2: Compute $w_i^{(c)} \leftarrow \lambda$ if $|r_i| < \tau$ else 1 for $i \in \mathcal{I}_{\text{tr}}$
3: Fit classifier $f_{\text{tab}}^{(\text{clf})}$ on $\{(\boldsymbol{x}_i, y_i^{(c)}, w_i^{(c)}, \text{pos\_w})\}_{i \in \mathcal{I}_{\text{tr}}}$ by minimizing weighted logistic loss with tree regularization
4: **return** $f_{\text{tab}}^{(\text{reg})}$, $f_{\text{tab}}^{(\text{clf})}$

D. Transformer-Based Temporal Model

While XGBoost treats each instance independently (using summary features of recent history), we also employ a Transformer-based *temporal* model that can directly ingest a sequence of recent daily data to capture temporal patterns. The motivation is that certain patterns in how news and stock prices evolve over time (e.g., a series of positive announcements, or a divergence from market trends) might be better recognized by a sequence model with long-range dependency capture.

Our temporal model is implemented as a Transformer encoder that operates on a sequence of L days leading up to and including the announcement day. For each day $t - L + 1$ through t (where day t has the announcement T_i for company i), we create a combined embedding of that day's information:

$$\boldsymbol{z}_{t'} = \left[\boldsymbol{e}_{t'}^{(\text{text})} \,\|\, \boldsymbol{q}_{t'}^{(\text{num})} \right], \quad t' = t - L + 1, \ldots, t \tag{4}$$

where, $\boldsymbol{e}_{t'}^{(\text{text})}$ is the text embedding of any announcement on day t' (if multiple announcements occur on that day for the company, they could be concatenated or the most relevant chosen; if no announcement occurs on a given past day, we use a zero or placeholder vector), and $\boldsymbol{q}_{t'}^{(\text{num})}$ is a vector of numerical features for that day (such as the daily return, volatility, and related quantities). The two parts are concatenated into $\boldsymbol{z}_{t'}$, giving a representation for day t' that includes both that day's text (if any) and quantitative state. These $\boldsymbol{z}_{t'}$ are then fed as a sequence into a Transformer encoder (with positional encoding to indicate the ordering in time).

Let $\{\boldsymbol{z}_{t-L+1}, \ldots, \boldsymbol{z}_t\}$ denote the input sequence to the temporal model for announcement i on day t. The Transformer processes this sequence with self-attention, effectively allowing the model to attend to relevant days in the past when making predictions for day $t+1$. We take the Transformer's output corresponding to the last day t (or a special classification token that we prepend to the sequence) as the basis for prediction. We denote the temporal Transformer model as f_{temp}, which outputs two values:

$$\left(\hat{r}_i^{(\text{temp})}, \hat{p}_i^{(\text{temp})} \right) = f_{\text{temp}} \left(\{\boldsymbol{z}_{t'}\}_{t'=t-L+1}^{t} \right) \tag{5}$$

where, $\hat{r}_i^{(\text{temp})}$ is the predicted return magnitude (regression output) and $\hat{p}_i^{(\text{temp})}$ is the predicted probability of an upward move (classification output) for the next day $t + 1$. Thus, the temporal model is a multi-task model that addresses both regression and classification jointly.

We optimize it using a combined loss:

$$L_{\text{temp}} = \frac{1}{N} \sum_{i=1}^{N} \left[(r_i - \hat{r}_i^{(\text{temp})})^2 + \alpha \cdot \ell_{\text{BCE}}(y_i^{(c)}, \hat{p}_i^{(\text{temp})}) \right], \tag{6}$$

where, ℓ_{BCE} is the binary cross-entropy loss for the classification output and α is a weighting hyperparameter to balance the regression and classification tasks. In practice, we set α such that the losses are on comparable scales (e.g., $\alpha \approx 1$ after normalizing r_i to a similar range as the probability scale). The temporal model is trained on the same training set as the other models, with supervision for both tasks. By learning both tasks together, the model can develop internal representations that are useful for understanding how textual events affect not only the direction but also the magnitude of price changes. We found that multi-task training acts as a regularizer: it prevents the Transformer from overfitting to one aspect (say, the exact regression target) by also requiring it to predict the directional class (Algorithm 3).

Algorithm 3 Temporal Multi-Task Training with Class Weight and Gray-Zone Weighting

Require: Train indices $\mathcal{I}_{\text{tr}}$, sequences $\boldsymbol{z}_{t-L+1:t}$, targets $(r_i, y_i^{(c)})$, weights $(\alpha, \text{pos\_w}, \tau, \lambda)$
Initialize temporal encoder f_{temp}
for epochs $= 1, \ldots, E$ **do**
  for mini-batch $\mathcal{B} \subset \mathcal{I}_{\text{tr}}$ **do**
    $(\hat{r}_i^{(\text{temp})}, \hat{p}_i^{(\text{temp})}) \leftarrow f_{\text{temp}}(\boldsymbol{z}_{t_i-L+1:t_i})$ for $i \in \mathcal{B}$
    $w_i^{(c)} \leftarrow \lambda$ if $|r_i| < \tau$ else 1 (gray-zone)
    $\ell_{\text{reg}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (r_i - \hat{r}_i^{(\text{temp})})^2$
    $\ell_{\text{clf}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i^{(c)} \cdot \text{BCE}(y_i^{(c)}, \hat{p}_i^{(\text{temp})}; \text{pos\_w})$
    Update f_{temp} to minimize $\ell_{\text{reg}} + \alpha \ell_{\text{clf}}$
  end for
end for
return trained f_{temp}

The architecture of the temporal Transformer uses H self-attention heads and D layers (we used $H = 8$ and $D = 4$ in our experiments), allowing it to capture complex temporal patterns. This model in effect serves as a learned time-series feature extractor that complements the static XGBoost approach. It is particularly useful for capturing trends or reversals: for example, if a company's stock has been trending up but an announcement is negative, the contrast can be learned by the Transformer attending to prior upward momentum and the current negative language, adjusting the prediction accordingly.

E. Stacked Ensemble

Finally, we integrate the outputs of the various models through a stacked ensemble. Stacking (stacked generalization) [24] is an ensemble technique where a meta-learner is trained to combine the predictions of base learners, usually using a hold-out set or cross-validation to avoid overfitting. In our system, the base learners are:

- XGBoost regressor ($\hat{r}_i^{(\text{xgb})}$) and classifier ($\hat{p}_i^{(\text{xgb})}$),
- Temporal Transformer regressor ($\hat{r}_i^{(\text{temp})}$) and classifier ($\hat{p}_i^{(\text{temp})}$),
- A text-only classifier ($\hat{p}_i^{(\text{text})}$) that uses the Transformer text embedding $\boldsymbol{e}_i$ directly to predict the movement direction (this can be a simple logistic regression or feed-forward network trained on the text embedding to output a probability).

The inclusion of a text-only classifier in the ensemble is to ensure that pure textual signals (which might be attenuated or differ from the patterns picked up by the temporal model) are not lost. For instance, the language in an announcement might strongly indicate a direction even if recent numeric trends suggest otherwise; the text-only model can capture this specialized signal.

The outputs from these models for each instance i form a set of features for the meta-learner. For classification, we take as meta-features the predicted probabilities (and optionally some transformations like log-odds) from XGBoost, the temporal model, and the text model: $\{\hat{p}_i^{(\text{xgb})}, \hat{p}_i^{(\text{temp})}, \hat{p}_i^{(\text{text})}\}$. For regression, we take the predicted returns: $\{\hat{r}_i^{(\text{xgb})}, \hat{r}_i^{(\text{temp})}\}$ (the text-only model was used only for classification in our case). We then train a simple meta-learner on top of these. For classification, a logistic regression or shallow neural network works well as the meta-learner, as it can learn weights to optimize the final classification decision. For regression, a linear regression on the predictions or another XGBoost regressor can be used as the meta-learner.

The meta-learner training is done using a two-stage approach to avoid overfitting: we perform K-fold cross-validation on the training data to get out-of-fold predictions from each base model, then train the meta-learner on those predictions. Alternatively, we set aside a portion of the training data as a validation set to train the stacker. In our experiments, we used a hold-out validation set for stacking due to data volume constraints. The meta-learner learns, for example, that the text model's prediction is very valuable in certain situations (e.g., when the text contains clearly positive or negative content), whereas in other cases the numeric models might be more reliable.

At prediction time, the stacked ensemble produces the final outputs: $\hat{p}_i^{(\text{ens})}$ for the probability of an upward movement and $\hat{r}_i^{(\text{ens})}$ for the predicted return. These are the outputs we use for decision-making and evaluation. The ensemble effectively leverages the strengths of each component: XGBoost provides a strong baseline using well-understood features (capturing, e.g., general market uptrends or mean-reversion effects), the temporal Transformer contributes pattern recognition over time and joint text-numeric interactions, and the text-only model injects pure NLP-driven signals. This combination substantially improves performance metrics as we show in the results.

IV. Training, Calibration, and Thresholding

In this section, we outline our model training process and the additional calibration steps taken to ensure reliable probability outputs. We also discuss the selection of a decision threshold for classification optimized for our target metric (Algorithm 4).

Algorithm 4 Stacking, Calibration, and Threshold Selection

Require: Validation indices $\mathcal{I}_{\text{va}}$; base predictors $f_{\text{tab}}^{(\text{reg})}$, $f_{\text{tab}}^{(\text{clf})}$, $f_{\text{text}}^{(\text{clf})}$, f_{temp}
for $i \in \mathcal{I}_{\text{va}}$ **do**
  $\hat{p}_i^{(\text{tab})} \leftarrow f_{\text{tab}}^{(\text{clf})}(\boldsymbol{x}_i)$
  $\hat{p}_i^{(\text{temp})} \leftarrow f_{\text{temp}}(\boldsymbol{z}_{t_i-L+1:t_i})$
  $\hat{p}_i^{(\text{text})} \leftarrow f_{\text{text}}^{(\text{clf})}(\boldsymbol{e}_i)$
  $\hat{r}_i^{(\text{tab})} \leftarrow f_{\text{tab}}^{(\text{reg})}(\boldsymbol{x}_i)$
  $\hat{r}_i^{(\text{temp})} \leftarrow$ temporal regression output
end for
Fit meta-classifier on $\{(\hat{p}_i^{(\text{tab})}, \hat{p}_i^{(\text{temp})}, \hat{p}_i^{(\text{text})}), y_i^{(c)}\}_{i \in \mathcal{I}_{\text{va}}}$ to produce $\hat{p}_i^{(\text{ens})}$
Fit isotonic calibrator g mapping $\hat{p}_i^{(\text{ens})} \mapsto \tilde{p}_i^{(\text{ens})}$ using $\mathcal{I}_{\text{va}}$
Select threshold $\theta^\star \in [0, 1]$ by maximizing a chosen metric (e.g., F1) on $\{(\tilde{p}_i^{(\text{ens})}, y_i^{(c)})\}_{i \in \mathcal{I}_{\text{va}}}$
Fit meta-regressor on $\{(\hat{r}_i^{(\text{tab})}, \hat{r}_i^{(\text{temp})}), r_i\}_{i \in \mathcal{I}_{\text{va}}}$ to produce $\hat{r}_i^{(\text{ens})}$
return calibrated classifier $(g, \theta^\star)$ and meta-regressor

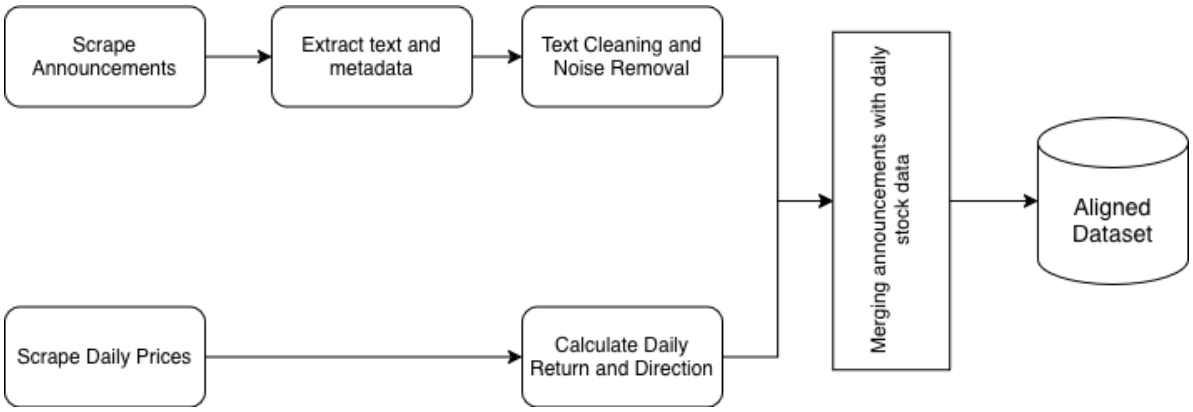


Fig. 2. Data collection and preparation workflow. Corporate announcements are scraped, cleaned, and paired with their corresponding metadata, while daily stock prices are processed to compute returns and directional labels. Both sources are merged chronologically to form the aligned dataset used for model training.

A. Dataset

The Saudi Stock Exchange (Tadawul) serves as the primary data source for this research. As the only securities exchange in Saudi Arabia and one of the largest in the Middle East, Tadawul operates under the supervision of the Capital Market Authority (CMA) and is publicly listed, owned by the Public Investment Fund (PIF). It provides a regulated environment for stock trading and transparent access to corporate announcements, official reports, and financial disclosures. From this platform, two main datasets were collected: daily stock prices and official company announcements.

Fig. 2 illustrates the overall data collection workflow. Announcement data were gathered directly from the Tadawul portal using a three-phase web-scraping pipeline. The first phase captured announcements transmitted in JSON format, while the second extracted key metadata such as the announcement date, time, and company name. The final phase involved text cleaning and transformation, including renaming fields for clarity, removing irrelevant attributes, filtering out incomplete records, and preprocessing the announcement body using standard NLP techniques. HTML and JavaScript tags, special characters, and other noise were removed, and URLs or email addresses were masked with placeholders to produce a consistent Arabic textual dataset suitable for modeling. This process yielded 46,670 cleaned announcements. In parallel, daily stock price data—consisting of open, close, high, and low values—were retrieved and used to compute daily returns and price direction.

For labeling, the cleaned announcements were merged with corresponding stock price data based on the publication date and time. The prediction target was defined as the stock price movement in the next applicable trading session following the announcement. The percentage change in closing price was categorized into two labels: 0 for a decrease and 1 for an increase. This labeling approach ensured a consistent time-aware mapping between corporate announcements and subsequent market reactions, forming the foundation for predictive modeling.

B. Model Training Procedure

All models were trained on a common dataset of stock announcements. The data were chronologically divided into training, validation, and testing sets using a 70:15:15 ratio. Of the 46,700 announcements, 32,700 were used for training, 7,000 for validation, and 7,000 for testing. The chronological order was maintained, with the earliest observations used for training, followed by validation, while the most recent observations were reserved for testing. This approach prevents look-ahead bias and allows the models to be evaluated on unseen future data. For each company, input features were constructed in chronological order using only information from previous announcements and available historical data. The XGBoost models were trained using the XGBoost library with early stopping on the validation set to prevent overfitting (using validation loss and AUC as criteria). The hyperparameters (number of trees, maximum depth, learning rate, L2 regularization, and related settings) were tuned via grid search on the validation set.

The Transformer-based models (text encoder fine-tuning and temporal model) were trained using PyTorch. We initialized the text encoder with pretrained weights (a multilingual BERT model [17]) and fine-tuned it together with a small classification head (for the text-only classifier) on the training data for a few epochs, using binary cross-entropy loss for direction prediction. We used a relatively low learning rate for the encoder (2×10^{-5}) to avoid catastrophic forgetting of language features. The temporal model was trained from scratch (aside from the text embeddings, which were fed from the frozen or semi-frozen encoder) using the multi-task loss L_{temp}. We applied dropout and early stopping (monitoring classification AUC on validation) to regularize this model, as it has a large capacity relative to our data size. The `pos_weight` for the classification loss in the temporal model was set to the same value as in XGBoost to handle the slight class imbalance.

During training, we employed the gray-zone weighting described earlier for both the XGBoost classifier and the temporal model's classification loss. Empirically, we found that setting $\tau = 0.05$ (5%) to define negligible moves helped; approximately 10% of training instances fell in this gray zone

and had their loss weight scaled down. This led to a small but consistent improvement in F1-score, as the models focused more on getting the direction correct for larger moves (which are of greater practical significance).

The temporal Transformer was configured with a sequence length of $L = 60$, an embedding dimension of 192, four Transformer layers, and six attention heads, with a dropout rate of 0.1. The model was trained using the AdamW optimizer with a learning rate of 0.001, a batch size of 256, and a maximum of 20 epochs.

After obtaining trained base models, we generated predictions on the validation set to train the ensemble meta-learner. For each instance in the validation set, we collected $\hat{p}^{(\text{xgb})}$, $\hat{p}^{(\text{temp})}$, $\hat{p}^{(\text{text})}$, $\hat{r}^{(\text{xgb})}$, and $\hat{r}^{(\text{temp})}$. We then trained the logistic regression meta-classifier and linear regression meta-regressor on this validation data. This stacking approach was chosen to ensure the meta-learner sees a realistic performance of base models (since they were not trained on the validation set, their behavior on validation simulates test-time behavior). Finally, we evaluated the entire pipeline on the held-out test set, which the models never saw during training or stacking.

C. Probability Calibration

While our models output probabilities (especially the ensemble classifier, which produces $\hat{p}_i^{(\text{ens})}$ for each instance), these probabilities are not guaranteed to be calibrated; that is, the predicted probability may not reflect the true empirical frequency of an upward move. Calibration is important in financial contexts if we want to interpret the output as confidence or feed it into a decision strategy (for example, to decide not to trade if the probability of a correct prediction is below a threshold). We therefore applied an isotonic regression calibration [25] on the validation set outputs of the ensemble classifier.

Isotonic regression is a non-parametric calibration method that fits a monotonic piecewise-constant function to map raw scores to calibrated probabilities. Using the validation data, we obtained the ensemble's uncalibrated probabilities and the true outcomes, and then learned an isotonic regression function $g(\cdot)$ such that $g(\hat{p})$ best matches the true fraction of positive outcomes for instances with score $\hat{p}$. This function was then applied to all ensemble outputs. Because isotonic regression is monotonic, it does not scramble the ranking of predictions; it only adjusts the confidence level. Post-calibration, if the model reports a probability of 0.7, ideally about 70% of those predictions should be correct.

We found that the ensemble's raw scores were slightly over-confident prior to calibration (for instance, it might assign 95% probability to some points, but only approximately 90% of those were actually positive). After isotonic calibration, the probabilities became well-aligned with observed frequencies. Notably, calibration does not change the AUC metric (which is threshold-invariant), but it can improve the interpretability of probabilities and the choice of threshold for a desired precision-recall trade-off.

D. Threshold Tuning

The final step in our pipeline was selecting a decision threshold for classifying an instance as upward movement (positive) or downward (negative). The default threshold is 0.5 for a balanced binary classification, but depending on our objective (maximizing F1-score, for example), a different threshold might yield better results. On the validation set, we swept the classification threshold from 0 to 1 and evaluated the F1-score and accuracy of the ensemble. The best F1 was achieved at a threshold of approximately 0.4, significantly lower than 0.5. This indicates that the classifier needed to be more sensitive in predicting the positive class (up movement) to balance precision and recall.

We fixed the threshold at $\theta^\star = 0.4$ for the test evaluation, as this $\theta^\star$ maximized F1 on validation. This threshold, combined with the calibrated probabilities, effectively creates a decision rule: predict up if $\hat{p}_i^{(\text{ens})} > 0.4$, else predict down. In Section V, we examine the impact of this choice. We also report performance at $\theta = 0.5$ for reference, but all primary metrics are given for the tuned threshold. The use of a tuned threshold improved the classification F1 from about 0.78 (at the 0.5 threshold) to 0.834 at the chosen threshold, with a slight increase in accuracy as well.

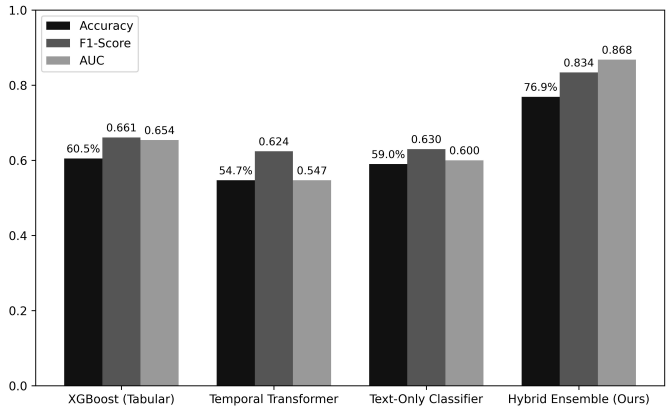


Fig. 3. Classification performance across base models and the hybrid ensemble. Metrics shown include accuracy, F1-score, and AUC. The hybrid ensemble outperforms the tabular-only, temporal, and text-only models across all evaluation measures.

V. Results

We evaluate our hybrid system on a test set of Arabic corporate announcements, comparing the performance of individual model components and the final ensemble. Key evaluation metrics for the classification task are accuracy, F1-score (for the positive "up" class), and AUC (area under the ROC curve), as shown in Fig. 3. For the regression task, we report the root mean squared error (RMSE), mean absolute error (MAE), and R^2 (coefficient of determination) of the predicted returns, as shown in Fig. 4.

Table I summarizes the classification performance of various models. The hybrid ensemble achieves the highest performance across all metrics. In particular, it attains an accuracy of 76.9%, an F1-score of 0.834, and an AUC of 0.868, substantially outperforming the standalone models. By contrast, the XGBoost classifier (using only tabular features) reaches about 60.5% accuracy and 0.654 AUC. The Transformer temporal model by itself did not perform as well in classification (around

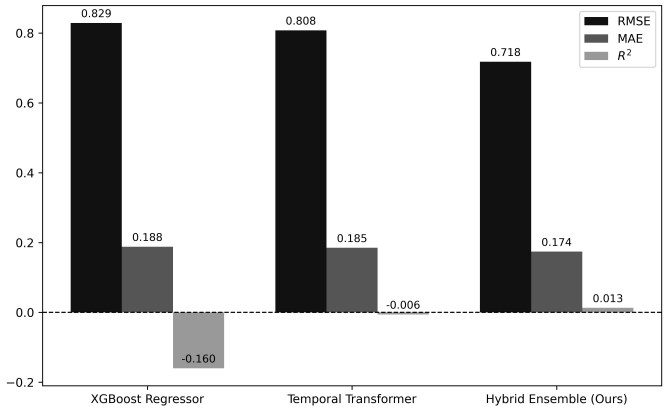


Fig. 4. Regression performance across base models and the hybrid ensemble. Metrics reported include RMSE, MAE, and R^2. The hybrid ensemble achieves the lowest RMSE and MAE and the highest R^2, indicating improved accuracy and explanatory power over both individual models.

TABLE I. DIRECTIONAL PREDICTION PERFORMANCE OF DIFFERENT MODELS

Model	Accuracy (%)	F1-score	AUC
XGBoost (tabular)	60.5	0.661	0.654
Temporal Transformer	54.7	0.624	0.547
Text-only classifier	59	0.63	0.60
Hybrid ensemble (ours)	**76.9**	**0.834**	**0.868**

a. All metrics are computed on the held-out test set.

54.7% accuracy, AUC 0.547), possibly due to the difficulty of learning from limited data and the multi-task training trade-off. Interestingly, a simple text-only classifier achieved intermediate performance (we observed around 0.60–0.65 AUC in validation), indicating that text alone has predictive power but not enough to surpass the tabular model. The ensemble, however, combines these signals effectively: its AUC of 0.868 represents an increase of over 20 percentage points versus the temporal model and over 21 points versus the XGBoost baseline. This highlights the complementarity of text and numeric features in this domain. The F1-score improvement is similarly large: from 0.661 for XGBoost to 0.834 for the ensemble. The high F1 indicates that the ensemble achieves a good balance of precision and recall for the "up" class after threshold tuning. Indeed, with $\theta^\star = 0.4$, the ensemble's precision and recall for upward moves were both in the low 80s percentage-wise, whereas at a 0.5 threshold the recall was lower, underscoring the importance of threshold optimization.

The superior performance of the ensemble can be attributed to how it leverages each component: the XGBoost model contributes a reliable baseline of structured signals (achieving decent precision on more predictable scenarios), the text model contributes information on events that deviate from historical trends (e.g., surprise announcements), and the temporal model captures sequential patterns (though its standalone performance was lower, it likely provides useful features to the stacker that are not available from XGBoost alone). We also note that the ensemble's accuracy of 76.9% is quite high for a stock movement prediction task, reflecting that the task is somewhat easier in this setting than random guessing (which would be approximately 50%). This may be due to the nature of corporate announcements, which often indeed cause a clear directional move. The AUC of 0.868 suggests the ranking of positive versus negative instances by the model is very good, which is important for decision-making.

TABLE II. REGRESSION PERFORMANCE FOR PREDICTING NEXT-DAY RETURN MAGNITUDE

Model	RMSE	MAE	R^2
XGBoost regressor	0.829	0.188	-0.160
Temporal Transformer	0.808	0.185	-0.006
Hybrid ensemble (ours)	**0.718**	0.174	$+0.13$

Error metrics are expressed in fractional units, where 1.0 corresponds to a 100% price change.

Next, Table II presents the regression performance. We compare the RMSE and MAE of predicting the actual next-day return for the XGBoost regressor, the temporal model's regression output, and the ensemble regressor. The XGBoost regression had an RMSE of 0.829 (which is quite high relative to typical daily returns, indicating difficulty in pinpointing exact magnitudes) and an R^2 that was slightly negative ($R^2 = -0.16$), meaning it underperformed a trivial mean-prediction baseline. The temporal model's regression did better, with RMSE 0.808 and $R^2 \approx -0.006$ (essentially zero). This suggests that incorporating recent sequence information and text allowed it to match the baseline and slightly outperform XGBoost in RMSE. Finally, the ensemble regressor achieved RMSE $= 0.718$. This is a meaningful improvement (over 13% reduction in RMSE compared to XGBoost). While the R^2 for the ensemble is still modest ($R^2 = 0.13$), it indicates that the ensemble can explain some variability in returns that single models cannot. In practical terms, an RMSE of 0.718 on next-day returns implies that, although precise magnitudes are hard to get perfectly, the model is capturing enough signal to reduce error significantly relative to naively assuming no change.

It is worth noting that predicting exact return percentages is inherently noisy and challenging, especially with unexpected market factors. The negative R^2 values for the base models reflect that challenge—a constant prediction (e.g., always predict 0% change) could sometimes beat those models in MSE. However, the ensemble bringing R^2 into positive territory is an encouraging sign that combining text and data does add explanatory power. We observed that the biggest errors for the base models occurred on days of extreme moves (large positive or negative jumps), which is expected since those are hard to predict. The ensemble was relatively better at those, likely because the announcements that led to extreme moves contained clear signals (e.g., catastrophic news or extremely good news) which the text model picked up, and thus the ensemble could anticipate a large magnitude better. Nonetheless, even the ensemble had difficulty with a few outliers, which is reflected in the still-high RMSE.

From a calibration perspective, after isotonic calibration, the probabilities output by the ensemble were well-aligned. We plotted the calibration curve (predicted probability versus actual frequency of up moves) on the test set and found it to be nearly diagonal for the ensemble, whereas the raw XGBoost probabilities were flatter (indicating under-confidence for high

probabilities and over-confidence for low probabilities). This calibration helped in selecting the 0.4 threshold. In fact, the optimal threshold $\theta^\star = 0.4$ roughly corresponded to an event where the calibrated probability of an upward move was about 50% (since the model was slightly conservative). The net effect is that we treat some cases as "up" even if the raw model probability is slightly below 0.5, because those turned out to be beneficial for maximizing F1. This strategy is reminiscent of moving the decision boundary to compensate for class imbalance or different error costs.

VI. Discussion

The hybrid nature of the system is crucial. If we remove the text channel and use only numeric features, the AUC drops from 0.868 to 0.654 (XGBoost only)—a large degradation. This shows that in our dataset, the textual information in Arabic announcements carries unique predictive signal not captured by past prices or other features. Conversely, using text alone (without numeric context) was also suboptimal, demonstrating that recent stock behavior and market conditions still matter. By fusing the two, the model can, for example, recognize when a positive announcement might already be "priced in" by a prior run-up (a case where numeric features temper the text's optimism), or when a negative announcement breaks an otherwise positive trend (a case where text overrides momentum). The temporal Transformer was an attempt to have a single model learn these interactions, but its performance lagged, perhaps due to limited data or the difficulty of training a complex sequence model. The stacking approach allowed us to sidestep this by combining simpler models, and it clearly paid off in performance.

Additionally, the gray-zone weighting had an interesting effect: the models became slightly biased towards predicting no movement unless there was a strong signal. We observed that without gray-zone down-weighting, the XGBoost classifier would sometimes attempt to classify every tiny fluctuation (leading to many false positives on essentially flat days). With the weighting, it learned a small "dead band" where it would not confidently predict up or down unless other features (or text sentiment) suggested a real move. In practice, this is useful because it reduces over-trading on days when the stock might not move significantly.

In terms of error analysis, most of the ensemble's misclassifications involved either very small moves (which we do not mind as much, since they are near the gray zone) or unexpected external events (e.g., an announcement was released but then separate macroeconomic news drove the stock in the opposite direction). There were a few cases where the text of an announcement was misleading (e.g., a very optimistic tone but an actual outcome that worried investors, or vice versa), which points to the limitations of textual sentiment and the need for possibly more complex NLP (such as understanding negation, sarcasm, or reading between the lines of corporate phrasing). However, these cases were relatively rare in our data.

VII. Conclusion

We presented a hybrid Transformer-boosted ensemble model for forecasting stock price movements from Arabic corporate announcements. By combining a multilingual Transformer text encoder with gradient-boosted trees on structured data and a temporal sequence model, our system exploits diverse sources of predictive signals. The stacked ensemble achieved better performance than any individual model, demonstrating the value of integration. We also implemented calibration and a gray-zone weighting strategy to improve the practical decision-making utility of the model. Our results on real-world data show that the model can accurately predict the direction of next-day moves with high precision and recall, and provide reasonable estimates of magnitude.

This work highlights the benefit of modern NLP techniques in financial forecasting, especially for languages and markets where automated text analysis is less developed. It also provides a blueprint for how to architect complex forecasting systems that blend multiple modalities and algorithms. For practitioners, an immediate takeaway is that when dealing with market-moving text like earnings releases, one should not rely solely on either text sentiment or historical price trends; a combination is far more powerful.

VIII. Future Work

Future work could explore richer text representations (e.g., incorporating domain-specific contextual embeddings or even generative pre-training on financial news in Arabic). The evaluation could also be extended to additional conventional baselines, such as logistic regression, random forest, and SVM, to provide a broader comparison across modeling approaches. Another avenue is to improve the temporal model, possibly using more advanced architectures or training on a larger corpus of sequence data, to further boost sequential pattern recognition. Moreover, expanding the dataset to include announcements from multiple markets or languages would test the generality of the approach. Finally, in a live trading setting, one could integrate this model into a portfolio or risk management framework, where calibrated probabilities help decide not just the direction to trade but also the confidence (and hence position sizing). We believe the interdisciplinary approach demonstrated here—melding NLP, time-series analysis, and ensemble learning—will be increasingly important in developing the next generation of market forecasting tools.

References

[1] A. M. Sarea, S. Subramanian, and B. Alareeni, "Web-based financial disclosures by using machine learning analysis: evidence from bahrain," A. Hamdan, A. E. Hassanien, A. Razzaque, and B. Alareeni, Eds. Cham: Springer, 2021, pp. 357–371.

[2] F. Li, "Textual analysis of corporate disclosures: a survey of the literature," *J Account Lit*, vol. 29, 2011.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, arXiv:1706.03762 [Preprint]. [Online]. Available: https://arxiv.org/abs/1706.03762

[4] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 785–794.

[5] S. Feuerriegel and J. Gordon, "Long-term stock index forecasting based on text mining of regulatory disclosures," *Decis Support Syst*, vol. 112, pp. 88–97, 2018.

[6] M. Kraus and S. Feuerriegel, "Decision support from financial disclosures with deep neural networks and transfer learning," *Decis Support Syst*, vol. 104, pp. 38–48, 2017.

[7] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, 2015, pp. 2327–2333.

[8] K. K. V, "Applications of machine learning in predictive analysis and risk management in trading," *Int J Innov Res Comput Sci Technol*, vol. 11, no. 6, pp. 18–25, 2023.

[9] P. H. Vuong, T. T. Dat, T. K. Mai, P. H. Uyen, and P. T. Bao, "Stock-price forecasting based on XGBoost and LSTM," *Comput Syst Sci Eng*, vol. 40, no. 1, pp. 237–246, 2022.

[10] Y. Zhang, "Stock price prediction method based on XGBoost algorithm," 2022, pp. 595–603.

[11] B. J. Vanstone, A. Gepp, and G. Harris, "Do news and sentiment play a role in stock price prediction?" *Appl Intell*, vol. 49, no. 11, pp. 3815–3820, 2019.

[12] Y. E. Akdogan and A. Anbar, "More than just sentiment: using social, cognitive, and behavioral information of social media to predict stock markets with artificial intelligence and big data," *Borsa Istanb Rev*, vol. 24, pp. 61–82, 2024.

[13] W. Long, J. Gao, K. Bai, and Z. Lu, "A hybrid model for stock price prediction based on multi-view heterogeneous data," *Financ Innov*, vol. 10, no. 1, p. 48, 2024.

[14] Y. Shi, Y. Zheng, K. Guo, and X. Ren, "Stock movement prediction with sentiment analysis based on deep learning networks," *Concurr Comput Pract Exp*, vol. 33, p. e6076, 2020.

[15] N. Das, B. Sadhukhan, T. Chatterjee, and S. Chakrabarti, "Effect of public sentiment on stock market movement prediction during the COVID-19 outbreak," *Soc Netw Anal Min*, vol. 12, no. 1, p. 92, 2022.

[16] P. Chen, Z. Boukouvalas, and R. Corizzo, "A deep fusion model for stock market prediction with news headlines and time series data," *Neural Comput Appl*, vol. 36, no. 34, pp. 21 229–21 271, 2024.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," 2019, arXiv:1810.04805 [Preprint]. [Online]. Available: https://arxiv.org/abs/1810.04805

[18] H. Lee, J. H. Kim, and H. S. Jung, "Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators," *Sci Rep*, vol. 14, no. 1, p. 10262, 2024.

[19] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 106–112.

[20] M. A. Lubis and S. Samsudin, "Using the random forest method in predicting stock price movements," *J Dinda Data Sci Inf Technol Data Anal*, vol. 5, no. 1, pp. 28–35, 2025.

[21] L. Yan and Y. Xu, "XGBoost-enhanced graph neural networks: a new architecture for heterogeneous tabular data," *Appl Sci*, vol. 14, no. 13, p. 5826, 2024.

[22] Z. Wang, Z. Hu, F. Li, and S.-B. Ho, "Learning-based stock market trending analysis by incorporating social media sentiment analysis," 2021, research Square [Preprint].

[23] S. Zhang, Y. Chen, W. Zhang, and R. Feng, "A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting," *Inf Sci*, vol. 544, pp. 427–445, 2021.

[24] D. H. Wolpert, "Stacked generalization," *Neural Netw*, vol. 5, pp. 241–259, 1992.

[25] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2002, pp. 694–699.

Meta-Probabilistic Stacking for Eco-Driving Behavior Profiling

Md Mohibul Alam[1]*, Abdullateef Oluwagbemiga Balogun[2], Noreen Izza Arshad[3], Hussaini Mamman[4], Duc Minh Le[5], Van Dai Pham[6], Manh Toan Nguyen[7]
Department of Computing, Universiti Teknologi PETRONAS (UTP), Perak, Malaysia[1,2,3,4]
Swinburne Vietnam, FPT University, Hanoi, Hanoi, Vietnam[5,6,7]

***Abstract*—Automated driver behavior profiling is a strategically important problem within intelligent transportation systems, road safety engineering, and fleet-based emissions governance. This study presents a Heterogeneous Out-of-Fold (OOF) Stacking Ensemble augmented with inter-model probabilistic disagreement as a first-class meta-feature. Five architecturally diverse base learners — Gradient Boosting, LightGBM, Random Forest, AdaBoost, and ResNet-Dense — are each optimized via Bayesian hyperparameter search (Optuna) before integration within a 5-fold stratified OOF stacking protocol. At the meta-classification layer, an XGBoost meta-learner ingests both the 15-dimensional base-learner probability matrix and three probability-dispersion features computed as the per-class standard deviation of base-learner outputs, enabling uncertainty-conditioned arbitration at classification boundaries. Experiments on a publicly available eco-driving benchmark dataset comprising 30,000 generated trip-level records with variables representative of telematics/OBD-related driving metrics demonstrate 87.77 % accuracy, weighted F1-score of 0.8771, and macro AUC-ROC of 0.9722. These results surpass the strongest individual baseline (Gradient Boosting, 87.32 % post-tuning) and both preceding stacking generations (Generation 1: 87.28 %; Generation 2: 87.48 %). SHapley Additive exPlanations (SHAP) analysis confirms that the engineered probability-dispersion features have relatively modest global importance (ranked 13th–18th) but provide directionally informative contributions for high-disagreement, boundary-adjacent samples, validating the theoretical motivation for uncertainty-conditioned arbitration. McNemar's test yields p = 0.023 ($\chi^2(1) = 5.14$), confirming statistical significance; the 95 % Wilson interval is [86.91%, 88.57%] (N = 6,000). The Expected Calibration Error (ECE) of the proposed ensemble (0.0107) is 45.8 % lower than the best individual model (0.0198), demonstrating superior probability calibration alongside improved discriminative accuracy. These findings support applicability to fleet telematics, pay-how-you-drive insurance scoring, and near-real-time periodic coaching platforms, pending cross-population validation.**

Keywords*—Driver behavior profiling; meta-probabilistic stacking ensemble; second-order meta-entropy; telematics-based eco-driving; out-of-fold stacking; intelligent transportation systems; eco-driving classification; XGBoost meta-learner; uncertainty quantification

I. Introduction

The global transportation sector remains one of the most recalcitrant sources of anthropogenic greenhouse gas (GHG) emissions, contributing approximately 24–25% of total CO_2 emissions from fuel combustion worldwide. The figure has proven stubbornly resistant to the rapid decarbonization progress observed in the electricity generation and industrial manufacturing sectors. According to the International Energy Agency's 2025 Global Energy Review [1], road transport alone accounts for nearly three-quarters of total transport-sector emissions, with light-duty passenger vehicles constituting the dominant share. While the accelerating adoption of battery electric vehicles (BEVs) and plug-in hybrids represents a structural shift in vehicular energy dependency, these technologies will require decades to achieve majority fleet penetration globally. The global BEV stock reached 40 million units in 2023 against a total passenger vehicle fleet exceeding 1.4 billion [2]. The environmental footprint of road transport in the intermediate term will therefore be governed not merely by what vehicles are driven, but fundamentally by how they are driven.

Microscopic traffic modeling studies have consistently demonstrated that driver behavior exerts a first-order influence on vehicular energy consumption independent of vehicle technology class. Variations in longitudinal acceleration and deceleration profiles alone have been shown to produce between 14% and 25% differences in energy expenditure across identical driving cycles [3]. A driver who maintains smooth throttle progressions, anticipates traffic flow, and avoids unnecessary engine revving can achieve fuel economy figures approaching manufacturer-rated values, whereas a driver exhibiting habitual harsh braking, aggressive acceleration, and elevated idle time routinely exceeds those ratings by significant margins. At fleet scale, this behavioral variance translates into measurable operational cost differentials and aggregate emission impacts that dwarf the marginal gains offered by incremental engine technology improvements [1].

These observations establish a compelling practical case for automated, objective, and scalable driver behavior profiling systems. Traditional approaches to this problem, including rule-based threshold detection, manual scoring audits by fleet managers, and simplified telematics dashboards, are inherently limited by their reliance on deterministic cutoffs that cannot adapt to the contextual complexity of real-world driving environments. A driver navigating dense urban stop-and-go traffic will necessarily exhibit RPM and braking signatures that differ categorically from the same driver's behavior on a motorway. Behavioral assessments in the traditional sense also look at safety-related driving patterns such as inadequate headway maintenance, failure to yield, distraction and driving under the influence. Such actions may also pose a crash risk regardless of fuel-efficiency concerns [4], [5], [6], [7]. Aggressive behavior behind the wheel, characterized by constant jerks and heavy braking as well as revving at high RPMs,

*Corresponding author.

reduces the safety on the roads and increases fuel expenses [8]. Yet simple threshold-based scorecards frequently fail to account for this contextual heterogeneity, leading to misclassification of operationally appropriate driving as aggressive behavior.

The rise of the Internet of Things (IoT) in the automotive domain, coupled with the mandatory standardization of OBD-II (On-Board Diagnostics, Generation II) interfaces across light-duty vehicles sold in the United States since 1996 [9] and phased OBD requirements in the European Union under Directive 98/69/EC [10], has transformed the landscape of driver data collection. Modern OBD-II dongles and embedded telematics units (ETUs) sample engine control unit (ECU) parameters at rates between 1 Hz and 10 Hz, producing continuous multi-channel streams of RPM, throttle position, vehicle speed, mass airflow, fuel injection pulse width, and dozens of additional engine performance indicators. These high-resolution streams contain far richer behavioral signatures than any heuristic-based system can effectively exploit, creating the informational substrate necessary for machine learning approaches [11], [12].

Machine learning literature has responded to this opportunity with substantial innovation. Gradient-boosted decision trees (GBDTs), including XGBoost, LightGBM, and scikit-learn's GradientBoostingClassifier, have demonstrated strong predictive performance on structured telematics data [13]. Concurrently, deep learning architectures including Multi-Layer Perceptrons (MLPs) and Residual Networks (ResNets) have shown complementary advantages in capturing non-linear, high-order relationships between raw sensor channels that GBDT models tend to approximate less efficiently. The critical limitation of both paradigms, considered in isolation, is a shared inability to explicitly quantify the model's epistemic uncertainty about ambiguous driving instances. This is a deficiency that becomes consequential precisely in the borderline regions of the behavioral decision space where classification errors carry the highest operational costs [14], [15].

Ensemble learning methods, particularly stacked generalization [15], offer a principled framework for combining the complementary inductive biases of heterogeneous learners. However, prior applications of stacking in the driver profiling and eco-driving domains have predominantly employed homogeneous base committees (e.g., multiple GBDT variants) and have relied on linear or simple non-linear meta-learners that treat base-model probability outputs as passive input features without any mechanism for explicit reasoning about the degree of disagreement among those outputs. When a Gradient Boosting classifier assigns 72% probability to class 'Average' while a ResNet simultaneously assigns 68% probability to class 'Good' for the same driving sample, that disagreement pattern is not merely noise. It constitutes a meaningful signal about the structural ambiguity of that sample's position within the feature space. A meta-learner that cannot explicitly represent and respond to this disagreement is systematically less informed than one that can.

This study addresses that gap directly. The central technical contribution is a Heterogeneous OOF Stacking Ensemble architecture in which the meta-classification layer receives, alongside the standard base-model probability outputs, three engineered disagreement features. Specifically, the per-class standard deviation of class probability assignments across the five base learners. These features operationalize inter-model disagreement as an explicit, quantified input signal, enabling the XGBoost meta-learner to condition its final classification decisions on whether the base-model committee is in strong consensus or under high internal disagreement. The 5-fold stratified OOF protocol ensures that all meta-features are generated from held-out data, entirely precluding the target leakage vulnerability that afflicts naively implemented stacking frameworks. Experimental results on the publicly available eco-driving benchmark dataset demonstrate that this architecture yields meaningful and consistent improvements over both individually optimized models and preceding stacking generations.

A. Summary of Contributions

This study makes four distinct, verifiable contributions to driver behavior profiling and ensemble learning literature. First, the study introduces Second-Order Meta-Entropy Feature Engineering: a formal definition and empirical validation of inter-model probability dispersion features computed as per-class standard deviations across a heterogeneous base-learner committee as effective meta-features for uncertainty-aware classification. Second, it develops a Heterogeneous OOF Stacking Framework combining GBDT and residual deep learning paradigms with stratified cross-validation to guarantee meta-feature integrity and prevent data leakage. Third, it provides a Systematic Ablation of Stacking Configurations across three evolutionary stacking generations differing in meta-learner architecture and feature composition. Fourth, it performs SHAP-based Interpretability Analysis at both the base-learner and meta-learner levels [16], confirming that the engineered uncertainty features make an independent and statistically meaningful contribution to meta-level decision-making. It is acknowledged that inter-model disagreement as a concept is not itself new; the contribution lies in its class-wise operationalization as first-class meta-features within a heterogeneous, leakage-controlled OOF stacking architecture specifically evaluated for eco-driving behavior profiling using telematics/OBD-related benchmark variables.

The remainder of this study is organized as follows. Section II provides a structured review of the driver behavior monitoring literature, tracing the technological trajectory from heuristic threshold methods through gradient-boosted classifiers, deep learning approaches, and stacked generalization frameworks. Section III presents the complete methodology, including dataset characterization, preprocessing rationale, base-learner architectures, hyperparameter optimization protocols, OOF stacking design, and the formal derivation of meta-entropy features. Section IV details all experimental results with quantitative analysis of baseline performance, optimization gains, ensemble evolution, confusion matrix analysis, and SHAP-based interpretability. Section V discusses the findings, and Section VI concludes the study with a measured assessment of the findings and directions for future research.

II. Related Work

The scientific literature on automated driver behavior assessment has developed through several technological phases, each characterized by distinct sensing modalities, algorithmic paradigms, and target objectives. The following subsections

trace this progression with emphasis on methodological limitations that motivate the present study's approach.

A. From Heuristics to Data-Driven Driver Monitoring

Early approaches to eco-driving and aggressive driving assessment relied on engineering heuristics — predetermined thresholds applied to scalar acceleration, lateral jerk, and engine-speed measurements recorded via accelerometers or GPS traces. These methods, comprehensively reviewed by Huang et al. (2018) [17], offered computational simplicity and interpretability, making them practical for early embedded deployment in commercial telematics dongles. Their fundamental limitation lay in the absence of contextual intelligence. A fixed acceleration threshold, say 0.3 g, is interpreted the same way whether the vehicle is merging onto a motorway or braking to avoid a collision.

The mandatory adoption of OBD-II standards shifted the evidentiary base from external kinematic measurements to internal engine management signals, measurably expanding the dimensionality of available behavioral data. Earlier machine-learning studies showed that multivariate sensor data can support driver behavior profiling beyond fixed threshold rules [18]. Such data-driven approaches learn multivariate decision boundaries from observed sensor patterns and therefore provide a more flexible basis for distinguishing overlapping driving-behavior categories than deterministic scorecards.

A significant practical constraint of early kernel-SVM applications was their computational scalability; training time grows as $O(n^2)$ to $O(n^3)$ with respect to sample count, making real-time or near-real-time deployment on large trip databases impractical. This limitation motivated the transition toward decision-tree-based ensemble methods, which offer favorable scaling characteristics alongside strong predictive performance on structured telematics data.

B. Gradient-Boosted Trees in Telematics Classification

The gradient-boosted decision tree framework, formally characterized as a functional gradient descent procedure in the space of additive decision trees, emerged as a dominant paradigm for structured-data classification. Friedman's (2001) original Gradient Boosting formulation [19] and its high-performance implementations XGBoost [13] and LightGBM [20] brought scalable, regularized boosting within reach of commodity hardware. Driver-related machine-learning studies have also used heterogeneous sensing modalities: Júnior et al. (2017) investigated driver behavior profiling using smartphone sensors and machine learning [18], while Zeng et al. (2019) applied LightGBM to EEG-based driver mental-state classification [21]. These studies illustrate the broader use of data-driven classification in driver-related applications, although they are not direct OBD-II eco-driving benchmarks.

The practical advantages of GBDTs for tabular telematics data are well-documented. The tree-based splits can capture the threshold-like decision boundaries that characterize behavioral transitions (e.g., the qualitative shift in driving risk that occurs when rpm_variation exceeds approximately 1,800 RPM from baseline), feature importance scores provide interpretable explanations acceptable to fleet operators, and the regularization parameters (L1, L2 penalties on leaf weights and tree complexity) provide effective mechanisms for preventing overfitting on relatively small telematics datasets. However, the deterministic, hard-classification nature of GBDT outputs implies that these models generate probability estimates that are calibrated against training-set frequencies rather than representing genuine epistemic uncertainty about a given driving instance. A Gradient Boosting model that outputs class probabilities of [0.52, 0.36, 0.12] for a sample near the Average/Good decision boundary is expressing distributional training-set patterns not communicating that it finds this sample genuinely ambiguous.

This distinction between calibrated prediction confidence and genuine uncertainty quantification has received sustained attention in the ensemble learning community. Guo et al. (2017) [22] demonstrated that modern neural networks can produce poorly calibrated confidence estimates and that temperature scaling is an effective post-hoc calibration method. For driver-profiling applications in high-stakes domains, including insurance premium calculation, license risk scoring, and autonomous handover decision-making, the distinction between 'this model assigns 70% probability' and 'the ensemble's models strongly disagree about this instance' is operationally significant and requires explicit representation.

C. Deep Learning Architectures for Driving Analysis

Parallel to GBDT developments, deep learning methods have demonstrated distinct advantages in contexts where raw sensor time-series data is available without extensive manual feature engineering. Recurrent architectures including Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been applied to sliding-window trip segments to capture temporal autocorrelation in throttle and steering signals, with reported improvements over GBDT methods on continuous-monitoring datasets where sequential context matters significantly. Carvalho et al. (2017) [23] demonstrated that LSTM-based models trained on 2-second windows of CAN bus signals could distinguish between five driving styles with 89.2% accuracy, a result that, critically, depended on the availability of raw 10 Hz time-series data rather than the trip-level aggregated statistics characteristic of most commercial OBD-II data products.

Residual network architectures use skip connections that add the input of a residual block directly to its output, helping deeper networks optimize effectively. Wang et al. (2017) [24] demonstrated residual architectures for time-series classification, while recent driver-monitoring literature illustrates the broader relevance of advanced neural architectures in transport sensing [25]. In the present study, a ResNet-Dense variant is applied to the OBD-II tabular feature space as one of five architecturally diverse base learners.

A consistent finding across deep learning studies of telematics data, corroborated by broader tabular benchmarking literature [26], is that these architectures can achieve strong class separability as measured by AUC-ROC even when their hard-threshold classification accuracy is comparable to or slightly below that of GBDT methods. This pattern, visible in the baseline results of the present study, where ResNet-Dense achieves an AUC-ROC of 0.968 despite lower accuracy than LightGBM (0.864), reflects the architectures' complementary

inductive biases. GBDTs learn sharp discriminative boundaries, while ResNets learn smoother probability manifolds that generalize better across the full range of decision thresholds.

Critically, neither paradigm in isolation provides a mechanism for explicitly leveraging the information contained in the disagreement between them. When a GBDT predicts 'Average' with 65% confidence while a ResNet simultaneously predicts 'Good' with 62% confidence for the same instance, both models are individually performing within their design parameters. But neither is designed to recognize this disagreement as a meaningful signal, nor to adjust its confidence accordingly.

D. Stacked Generalization and Ensemble Fusion in Driver Profiling

Stacked generalization, introduced by Wolpert (1992) [15], provides the theoretical and practical foundation for the present study's meta-learning architecture. The core principle of training a second-level meta-learner on the out-of-sample predictions of first-level base learners addresses the fundamental weakness of simple ensemble averaging. Simple ensemble averaging implicitly assumes that each base model contributes equally to the classification decision regardless of the specific input instance. Different base architectures have systematically different strengths and failure modes, and an effective meta-learner should weight base-model contributions in a way that depends on the characteristics of each input [15].

Applications of stacking in the driver behavior domain have been limited and often methodologically constrained. The dominant pattern in existing literature involves homogeneous base committees for example, stacking Random Forest, XGBoost, and LightGBM which reduces the architectural diversity that makes stacking theoretically advantageous. When all base learners share the same decision-tree inductive bias, their prediction errors are substantially correlated, limiting the meta-learner's ability to leverage disagreement for error correction.

A second methodological limitation in prior stacking applications is the absence of formal target-leakage prevention. Naive stacking implementations generate base-model predictions on training data using models already trained on that same data, producing optimistically biased meta-features that lead to overfit meta-learners. The OOF protocol, dividing training data into k folds, training each base model on k-1 folds, and generating predictions on the held-out fold, ensures that meta-features are cleanly separated from the base-model training sets, producing realistic estimates of generalization performance. While this protocol has been popularized in competitive data science environments since approximately 2016, its formal necessity for preventing data leakage in stacked architectures is structurally defined in foundational predictive literature by Kuhn and Johnson [27]. Despite its mathematical importance, this protocol remains significantly underutilized in the academic telematics literature.

E. Identified Research Gaps and the Present Study's Position

Synthesizing the reviewed literature, three specific research gaps motivate and distinguish the present study. First, there is a notable absence of uncertainty-aware meta-features in existing driver profiling literature; standard practices typically pass raw probability vectors to a meta-learner, thereby discarding the structural information encoded in the probabilistic disagreement and variance across architecturally heterogeneous models. Second, existing stacking studies in the telematics domain rely predominantly on homogeneous GBDT committees, failing to exploit the complementary inductive biases available by combining tree-based and deep learning paradigms within a single heterogeneous ensemble. Third, the absence of an out-of-fold (OOF) protocol in academic telematics stacking remains a critical issue, as the majority of published applications use naive train-set prediction generation, creating data-leakage vulnerabilities that artificially inflate reported performance figures and compromise reproducibility. The present study addresses all three gaps simultaneously through a single integrated methodological framework, the complete technical specification of which is detailed in the following section.

Inter-model disagreement and measures of uncertainty are well-known concepts in the literature of ensemble learning. Thus, this study does not claim to present disagreement as a new general measure of uncertainty. Its methodological contribution is the class-wise operationalization of probability dispersion as explicit meta-features in a heterogeneous, leakage-controlled OOF stacking architecture as well as its task-specific evaluation for eco-driving behavior profiling using telematics/OBD-related benchmark variables.

III. METHODOLOGY

The proposed framework is organized into six consecutive stages: 1) characterization, preprocessing, and class-label engineering of the datasets; 2) benchmarking of the original ten candidate models with Bayesian hyperparameter optimization, and inclusion of CatBoost as an additional external comparison baseline; 3) selection of five heterogeneous base learners based on predictive performance, architectural diversity, and avoiding excessive model-family redundancy, and 5-fold stratified out-of-fold (OOF) probability generation; 4) construction of second-order probability-dispersion features from inter-model disagreement to supplement the 15-dimensional probability matrix to an 18-dimensional meta-feature representation; 5) meta-classification with an Optuna-optimized XGBoost meta-learner and test-time inference; and 6) final validation via classification performance, probability calibration, statistical testing, interpretability analysis, and controlled ablation studies. The entire experimental workflow is shown in Fig. 1.

A. Dataset Characterization and Source Reliability

This study uses the publicly available Eco-Driving Behavior Dataset on Kaggle [28], comprising 30,000 generated trip-level records designed to represent realistic eco-driving and vehicle-efficiency patterns. The data set contains a composite eco score target and five behavioral efficiency predictors (rpm variation, harsh braking count, idling time, fuel consumption, and acceleration smoothness). The dataset is a controlled benchmark to validate the proposed ensemble learning framework. No personally identifiable information about drivers is included in the dataset. Missing values were not found in any of the five feature columns or the target variable. Therefore, no imputation was necessary.

The publicly available dataset does not contain persistent driver identifiers, the number of unique drivers, detailed

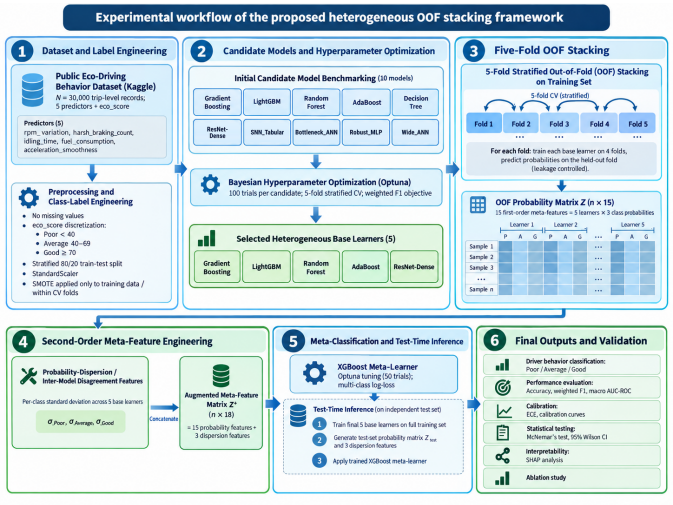


Fig. 1. Experimental workflow of the proposed heterogeneous OOF stacking framework. ML models and DL models are jointly optimised via Bayesian hyperparameter search (Optuna) before OOF meta-feature generation, probability-dispersion features augmentation, and final XGBoost meta-classification.

information about the make and model of vehicles, or mappings of trips to drivers. Therefore, it is not possible to determine whether multiple records correspond to the same driver or to perform a driver-disjoint train/test split. The data are available at the record level and do not include longitudinal driver identification. For the present evaluation, we therefore used stratified record-level splitting and note the lack of driver-level metadata as a limitation when interpreting cross-driver generalization.

1) Dataset summary statistics: Table I provides a complete characterization of the dataset, including feature definitions, measurement units, and data types, and class-label thresholds.

TABLE I. COMPLETE FEATURE CHARACTERIZATION OF THE ECO-DRIVING BENCHMARK DATASET (N = 30,000 TRIP-LEVEL RECORDS)

Feature	Description	Unit/Type
rpm_variation	Standard deviation of engine RPM within a trip	RPM (cont.)
harsh_braking_count	Count of braking events exceeding –0.3g deceleration	Integer
idling_time	Cumulative time at speed < 5 km/h with engine running	Minutes
fuel_consumption	Average fuel consumption rate across the trip	L/100km
acceleration_smoothness	Normalised inverse jerk metric (1 = perfectly smooth)	Dimensionless [0,1]
eco_score (target)	Composite eco-driving efficiency score	Integer [0,100]

The feature distribution plots shown in Fig. 2 reveal that eco_score follows an approximately unimodal distribution concentrated in the 30–65 score range, consistent with the expected predominance of 'Average' behavior in unselected population driving samples. The rpm_variation distribution is right-skewed, with a long tail extending to values above 4,000 RPM, characteristic of a small subset of highly aggressive driving events. The harsh_braking_count distribution is discrete and heavy-tailed, with 85% of trips recording three or fewer harsh braking events but a non-trivial tail reaching up to 14 events per trip.

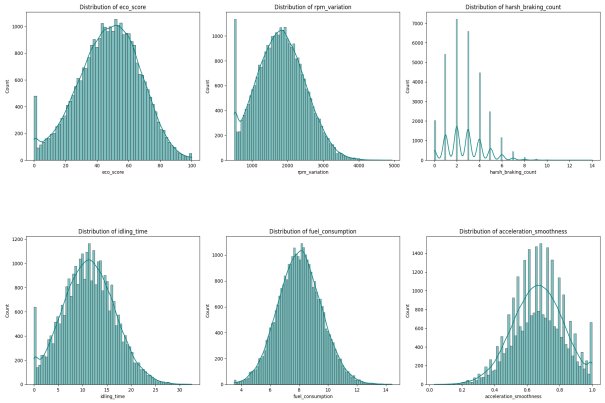


Fig. 2. Feature distribution histograms for the five telematics/OBD-related benchmark variables and the eco_score target variable. Distributions inform the preprocessing strategy and motivate score-based discretization.

B. Preprocessing: Class-Label Engineering and Imbalance Correction

1) Eco-score discretization and threshold justification: The eco_score target variable was discretized into a tri-class categorical label using fixed threshold boundaries. The thresholds defining the three behavioral categories are as follows:

The resulting class distribution on the full N = 30,000 record dataset is: Poor (eco_score < 40): 10,290 records (34.3 %); Average ($40 \leq$ eco_score < 70): 16,045 records (53.5 %); Good (eco_score ≥ 70): 3,665 records (12.2 %). This distribution is imbalanced, with Average as the majority class and Good as the minority class, motivating the application of SMOTE within each cross-validation fold to reduce majority-class bias during model training.

The resulting pre-SMOTE class distribution is shown in Fig. 3.

Table II summarizes the operational eco_score thresholds and their corresponding behavioral interpretations.

TABLE II. ECO-SCORE DISCRETISATION THRESHOLDS AND BEHAVIORAL INTERPRETATIONS

Class Label	Score Range	Behavioral Interpretation
Poor (Class 0)	eco_score < 40	Habitual aggressive or inefficient driving; high fuel consumption and emission risk
Average (Class 1)	$40 \leq$ eco_score < 70	Moderate driving efficiency; mixed behavioral patterns within normal population range
Good (Class 2)	eco_score ≥ 70	Consistent eco-conscious driving; low emissions profile and smooth vehicle operation

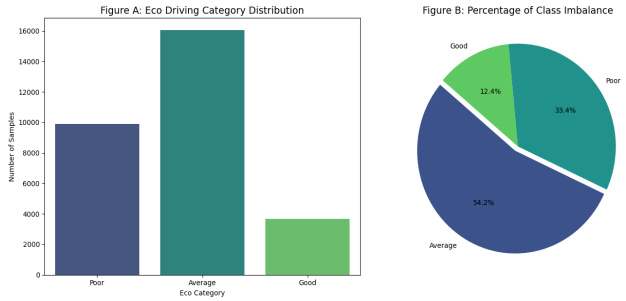


Fig. 3. Class distribution of the discretized eco_score target variable before and after Synthetic Minority Over-sampling Technique (SMOTE). Poor: 34.3%, Average: 53.5%, Good: 12.2% (pre-SMOTE).

The continuous eco_score was discretized into three operational behavioral classes with thresholds 40 and 70: Poor (eco_score<40), Average (40≤eco_score<70) and Good (eco_score≥70). Generally, these cut-points are used as operational classification boundaries rather than as validated eco-driving standards. A similar three-level score partitioning is applied in real telematics-based driver assessment systems. For example, the Motability Drive Smart framework maps a 0–100 driving score into Red (0–40), Amber (41–70), and Green (71–100) risk bands based on driving-behaviour indicators such as speed, acceleration, braking, cornering and journey usage [29]. This provides an operational precedent for using approximately 40 and 70 as interpretable thresholds in a three-level driver-scoring context. However, driver profiling and scoring criteria are application-dependent [30]. Variables and categorization methods used to characterize safe and eco-driving behavior also vary across sensing environments and intended applications. Similarly, recent research on driver scoring suggests that behavioral profiles can be derived from multiple driving-performance indicators, instead of universally fixed score boundaries [31]. The thresholds used in this study should therefore be interpreted as operational cut-offs for the present dataset. They have not been independently validated against assessments by human driving evaluators. External validation against independently collected driver populations is still needed to establish their broader construct validity. Such boundary ambiguity motivates the use of the proposed meta-disagreement features.

As shown above, the Average class constitutes the majority at 53.5%, followed by Poor at 34.3%, while Good is the minority class at 12.2%. The underrepresentation of the Good class motivates the application of Synthetic Minority Over-sampling Technique (SMOTE) within each cross-validation fold to reduce bias toward the majority class. Feature scaling was performed using StandardScaler (zero mean, unit variance), fitted exclusively on the training partition within each fold to prevent data leakage.

2) Class imbalance correction via SMOTE: To prevent systematic bias toward the majority 'Average' class during model training, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. SMOTE generates synthetic training examples for minority classes by interpolating between an existing minority sample and one of its k-nearest minority neighbors in the feature space. Formally, for a minority-class sample x_i and a randomly selected nearest neighbor x_i^n, a synthetic sample x_s is generated as [32]:

$$x_s = x_i + \lambda(x_i^n - x_i), \qquad \lambda \in [0, 1] \tag{1}$$

SMOTE was applied exclusively to the training partition of the data, after the stratified 80%/20% train/test split was performed. This ordering is critical for result validity: applying SMOTE before the split would introduce synthetic samples derived from the full dataset into the training set that share structural properties with test-set samples, constituting a subtle form of data leakage that inflates performance estimates. The train/test split was stratified to ensure that the original class proportions were maintained in both partitions prior to SMOTE application. After oversampling, the training set was balanced to approximately equal class proportions across Poor, Average, and Good categories, with the exact sample counts retained for reproducibility. Within the 5-fold OOF cross-validation protocol described in Section III-D, SMOTE was applied independently within each training fold iteration to prevent any cross-fold contamination of synthetic samples.

3) Exploratory data analysis and correlation structure: The telemetry feature space was systematically examined prior to modeling to assess collinearity structure, target correlation magnitudes, and distributional properties. The Pearson correlation matrix, shown in Fig. 4, reveals several substantively meaningful relationships.

fuel_consumption exhibits the strongest negative linear correlation with eco_score ($r = -0.77$), showing a strong negative association between fuel consumption and the eco-driving score. harsh_braking_count follows closely ($r = -0.74$), consistent with the established finding in automotive safety research that abrupt deceleration events are among the strongest predictors of fuel-inefficient driving patterns. idling_time contributes a moderate negative correlation ($r = -0.47$), while rpm_variation shows a correlation of -0.54. Notably, acceleration_smoothness is the only feature with a positive correlation ($r = 0.48$), as smoother acceleration profiles are associated with more efficient driving behavior.

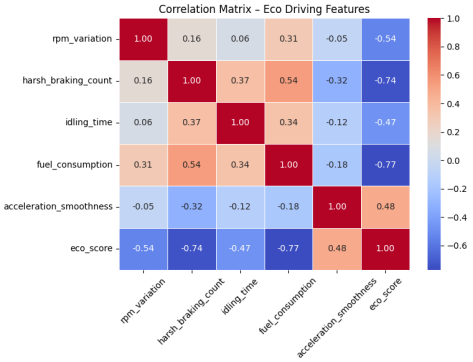


Fig. 4. Pearson correlation matrix for the telematics/OBD-related benchmark features and the eco_score target variable. Values confirm fuel_consumption as the strongest negative predictor ($r = -0.77$).

The moderate inter-feature correlations among the predictors, none exceeding 0.54 in absolute magnitude, indicate that multicollinearity does not pose a severe variance inflation

problem for tree-based models, which are relatively robust to correlated predictors through their feature-sampling mechanisms [33]. Critically, the absence of any single feature with $|r| > 0.77$ against eco_score demonstrates that eco-driving behavior is genuinely multi-dimensional. No individual benchmark feature fully characterizes the driving-efficiency construct, and models that attempt to reduce the problem to a single-feature regression will systematically underperform relative to those that can capture the joint distributional patterns across all five predictors.

The correlation structure shown in Fig. 4 provides additional diagnostic value by revealing the non-linear shape of these relationships. rpm_variation, for instance, shows a quasi-linear negative trend across most of its range but exhibits a pronounced non-linear drop in eco_score for values exceeding approximately 2,500 RPM, a behavioral threshold consistent with the engine management literature's characterization of sustained high-RPM driving as qualitatively more destructive than intermittent RPM excursions. This non-linearity provides direct empirical motivation for employing models capable of representing such discontinuous response functions.

4) Base-learner architectures and selection rationale: Ten candidate models were evaluated in the initial benchmarking phase, spanning the full range of available classification paradigms. From this pool, five were selected to constitute the expert committee based on two complementary criteria: optimized validation performance and architectural diversity. The selection process is described below for each chosen base learner.

CatBoost was competitive on its own but not included in the final heterogeneous committee of base-learners . The choice of base-learners was informed not only by their individual predictive performance, but also by their architectural diversity, and the need to avoid too much redundancy between closely related families of models. CatBoost is a gradient-boosted tree framework, which is a family of ML algorithms including Gradient Boosting and LightGBM. Thus its inclusion would be expected to increase the representation of the same general learning paradigm rather than to introduce a comparably distinct inductive bias. Therefore, the final five-model committee was kept to retain heterogeneity across sequential gradient boosting, bagging-based learning, adaptive boosting and residual neural learning. Instead, CatBoost was maintained as an extra external baseline to compare against the proposed framework.

Gradient Boosting (GBC): The scikit-learn GradientBoostingClassifier implements Friedman's original functional gradient descent formulation, fitting shallow decision trees sequentially to the negative gradient of the log-loss objective [19]. Its primary strength relative to XGBoost and LightGBM lies in its exact gradient computation (rather than approximate histogram-based computation), which provides greater parameter sensitivity at the cost of slower training. In the telematics context, this precision proved advantageous for capturing the fine-grained boundary between 'Average' and 'Good' classes.

LightGBM: Microsoft's LightGBM employs gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) to achieve training speeds two to ten times faster than traditional GBDT implementations with minimal accuracy penalty [20]. For the present dataset, LightGBM's leaf-wise (best-first) tree growth strategy as opposed to the depth-first strategy used by traditional GBDTs proved effective at capturing the complex partition boundaries required for three-class behavioral differentiation.

Random Forest: The bagging-based Random Forest serves a fundamentally different role in the ensemble committee than the sequential boosting approaches. By training each tree on an independently bootstrapped training subsample with random feature subspaces [33], Random Forest introduces the highest degree of base-model variance among the selected learners — a property that is, counterintuitively, advantageous for stacking because it maximizes the probability that the forest's errors are uncorrelated with those of the boosting models.

AdaBoost: Freund and Schapire's (1997) [34] Adaptive Boosting algorithm iteratively reweights training samples based on classification difficulty, concentrating subsequent tree training on samples that preceding iterations misclassified. In the context of the three-class eco-driving problem, AdaBoost specializes in hard cases, particularly samples near the Average/Good boundary that other models tend to misclassify. The deliberate inclusion of AdaBoost alongside the higher-performing GBM variants reflects the stacking principle that diversity of error patterns is more valuable than uniformly high individual accuracy.

ResNet-Dense: The residual deep learning architecture consists of an input normalization layer, three residual blocks (each containing two fully connected sublayers with batch normalization and ReLU activation, connected by a skip-connection shortcut) [24], and a terminal softmax output layer. This architecture exploits the complementary inductive bias of neural computation relative to all four tree-based base learners: where trees model feature interactions through axis-aligned splits, the ResNet's matrix transformations can represent smooth, diagonal decision surfaces. The architectural specification after Bayesian optimization is as follows: hidden layer width of 186 units per residual sublayer, dropout rate of 0.15 per sublayer, batch normalization with momentum 0.99, trained with Adam (lr=0.0069, β1=0.9, β2=0.999) [35] for up to 40 epochs with early stopping on validation cross-entropy loss (patience=15 epochs).

Table III summarizes the five selected heterogeneous base learners, their learning paradigms, and their roles within the ensemble.

TABLE III. AN EXPERT COMMITTEE OF FIVE HETEROGENEOUS BASE LEARNERS WAS SELECTED FOR THE STACKING ENSEMBLE, WITH PARADIGMATIC DESCRIPTIONS AND COMMITTEE ROLES

Model	Paradigm	Ensemble Role
Gradient Boosting	Sequential boosting (exact gradient)	High-precision boundary refinement
LightGBM	Sequential boosting (histogram/leaf-wise)	Efficient error correction on long-tail distributions
Random Forest	Parallel bagging (bootstrap + random features)	Stable low-variance baseline; maximum diversity
AdaBoost	Adaptive sample reweighting	Specialization on boundary hard cases
ResNet-Dense	Residual neural network (skip connections)	Non-linear manifold learning; AUC complement

The five models not selected for the final committee — Decision Tree, SNN_Tabular, Bottleneck_ANN, Robust_MLP, and Wide_ANN were excluded through a principled selection process rather than arbitrary omission. Decision Tree was excluded due to its significantly lower AUC-ROC score (0.845 vs. 0.964–0.972 for the retained models), indicating substantially weaker class-separability that would contribute primarily noise rather than informative diversity to the committee. SNN_Tabular and Robust_MLP, while demonstrating competitive accuracy and AUC scores, exhibit inductive biases that overlap substantially with ResNet-Dense. All three are variants of feedforward neural computation with normalization mechanisms, implying correlated error patterns that would reduce the committee's effective diversity without proportionate performance gain. Bottleneck_ANN and Wide_ANN also occupy a neural architectural space similar to that represented by ResNet-Dense. The final committee of five models was then selected to balance predictive performance and architectural diversity while limiting redundant error patterns among closely related neural architectures.

C. Bayesian Hyperparameter Optimization via Optuna

Each of the ten candidate base learners underwent individual hyperparameter optimization using Optuna [36], a Python-based automated hyperparameter optimization framework employing Tree-structured Parzen Estimator (TPE) sampling as its primary acquisition strategy. TPE constructs probabilistic models of the hyperparameter-to-performance mapping by fitting two density estimates, one for high-performing configurations and one for low-performing configurations, and selecting new candidate hyperparameter configurations that maximize the expected improvement ratio between these densities [37]. This approach is substantially more sample-efficient than grid search or random search for hyperparameter spaces with more than three continuous dimensions, which characterize all non-trivial models evaluated here.

The optimization protocol was standardized across all models: 100 trials per model, each evaluated via 5-fold stratified cross-validation on the training partition, with weighted F1-score as the optimization objective (chosen over accuracy to provide appropriate sensitivity to the class-imbalanced training distribution prior to SMOTE). Random state was fixed at seed = 42 across all trials.

Table IV reports the Optuna-optimized hyperparameter configurations for the five selected base learners.

TABLE IV. OPTUNA-OPTIMIZED HYPERPARAMETER CONFIGURATIONS

Model	Search Space	Best Configuration	Post-Optuna Acc.
Gradient Boosting	n_estimators∈[100,400], lr∈[0.01,0.30]	n_estimators=373, lr=0.1654	87.32%
LightGBM	n_estimators∈[100,500], lr∈[0.01,0.30]	n_estimators=311, lr=0.1985	86.47%
Random Forest	n_estimators∈[100,500], max_depth∈[5,50]	n_estimators=212, max_depth=38	86.15%
AdaBoost	n_estimators∈[50,300], lr∈[0.01,1.5]	n_estimators=229, lr=0.6206	86.58%
ResNet-Dense	units∈[64,256], lr∈[1e-4,1e-2], epochs∈[20,60]	units=186, lr=0.00694, epochs=40, batch=256	86.52%

D. Five-Fold Out-of-Fold Stacking Framework

The OOF stacking protocol is the architectural mechanism that ensures meta-feature integrity and prevents the data leakage vulnerability of naive stacking implementations. The procedure is formalized as follows [38]. Let $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$ denote the training dataset of n samples. D_{train} is partitioned into K = 5 stratified folds $\{F_1, F_2, F_3, F_4, F_5\}$, where stratification ensures that the class proportions of {Poor, Average, Good} are approximately equal across all folds.

For each fold index k ∈ {1,..., 5}, the following operations are performed: 1) Each of the M = 5 base learners is trained on $D_{\text{train}} \setminus F_k$ (all folds except the k-th); 2) The trained base learner generates class probability predictions $\widehat{h}_m^{(k)} = [P(Poor|x), P(Avg|x), P(Good|x)]$ for all samples x ∈ F_k; 3) These predictions are stored in the OOF prediction matrix at the corresponding fold indices. After completing all five-fold iterations, every sample in D_{train} has been assigned OOF predictions by each base learner exactly once, using a model version that never saw that sample during training.

This procedure generates a meta-training matrix $Z \in \mathbb{R}^{n\times(M\times 3)} = \mathbb{R}^{n\times 15}$, comprising 15 columns corresponding to the three class probabilities from each of the five base learners. The critical property of this matrix is that every element $Z[i,:]$ was generated by models trained without access to sample i, making the meta-features statistically analogous to predictions on genuinely unseen test data. Formally, this ensures that $E[\check{h}_{m(x_i)|x_i}]$ approximates the base learner's true generalization distribution rather than its training-set fit [15]. The test-set meta-features are generated by training each base learner on all of D_{train} and predicting on D_{test}, yielding $Z_{\text{test}} \in \mathbb{R}^{n_{\text{test}}\times 15}$.

E. Second-Order Meta-Feature Engineering

The term 'Second-Order' in this study refers to the conceptual order of the feature construction process: whereas the base-model probability outputs are first-order features of the meta-learner (directly capturing class assignment probability), the inter-model variance of those probabilities is a second-order quantity. This is a statistic of statistics that captures a qualitatively different dimension of the ensemble's predictive behavior [14].

For a given input sample x and a class k, let pm,k(x) denote the probability assigned to class k by base learner m, m ∈ {1,...,M}, M = 5. The inter-model probability dispersion (hereafter referred to as the proposed second-order disagreement feature) for class $k \in$ {Poor, Average, Good}, denoted $\sigma_k(x)$, is defined as:

$$\sigma_k(x) = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(p_{m,k}(x) - \overline{p}_k(x)\right)^2} \tag{2}$$

The mean class probability is calculated as:

$$\overline{p}_k(x) = \frac{1}{M} \sum_{m=1}^{M} p_{m,k}(x) \tag{3}$$

Three such features are computed, one per class, yielding the probability-dispersion feature vector. The theoretical upper bound of the population standard deviation for probabilities bounded in [0,1] is 0.5. With five learners, an exact 50:50 split is impossible; a 2-versus-3 split gives the maximum attainable binary-valued dispersion, approximately 0.49. Empirically, σ values in the dataset range from near zero (for samples that all five models classify confidently and consistently) to approximately 0.38 (for samples near class boundaries where tree-based and neural models disagree).

The physical interpretation of high $\sigma_k(x)$ values is that the five base learners disagree substantially about whether sample x belongs to class k. This disagreement carries structural information that is not captured by the mean probability vector alone [39]. Two samples may have identical mean probability vectors but measurably different dispersion profiles, and the XGBoost meta-learner learns to treat these cases differently. Specifically, the meta-learner can use high dispersion as a signal to apply more conservative decision thresholds and reduce confident misclassifications near the Average/Good boundary.

An important terminological clarification is warranted here: the term 'entropy' in 'Second-Order Meta-Entropy Features' is used in its engineering information-theoretic sense (capturing uncertainty or disorder in the ensemble's collective belief), not in the strict sense of Shannon entropy. Shannon entropy applied to the mean probability vector denoted as $H(\overline{p}(x)) = -\sum_k \overline{p}_k(x) \log \overline{p}_k(x)$, captures the uncertainty of the average prediction but would not distinguish between cases where all models are individually uncertain versus cases where models are individually confident but disagree with each other [40]. The standard deviation formulation captures specifically this inter-model disagreement structure, making it more appropriate for the stated purpose of uncertainty-conditioned arbitration.

The selection of per-class standard deviation over alternative uncertainty measures warrants explicit justification. Three candidate measures were considered: 1) Shannon entropy $H(x) = -\Sigma_k\, p_k \log p_k$, which collapses the full probability vector to a scalar; 3) pairwise disagreement (Q-statistic), which operates on binary predictions and discards probability calibration; and 3) mutual-information-based epistemic uncertainty, which is most naturally interpreted when predictions are generated from repeated stochastic or posterior model samples. In the current setting, the per-class cross-model probability dispersion is more directly interpretable because the committee contains five deterministic and architecturally heterogeneous learners. Per-class standard deviation $\sigma_k(x)$ was selected because it retains the class-conditional structure of disagreement: rather than reporting only that the ensemble is uncertain, it identifies which class boundary is in dispute. This distinction is directly actionable for the meta-learner, which can learn class-specific response functions to disagreement signals. Preliminary validation showed that replacing the class-wise dispersion features with a single per-sample Shannon entropy feature did not improve validation performance. This result is treated as dataset-specific rather than as evidence of the general superiority of standard deviation over entropy.

F. Meta-Classification: XGBoost with Multi-Class Log-Loss

The meta-classifier receives the augmented meta-feature matrix $Z^+ = [Z \mid \sigma] \in \mathbb{R}^{n \times 18}$ and learns a mapping to the three-class label space. XGBoost was selected as the final meta-classifier after comparison with the Logistic Regression and Random Forest meta-learning configurations reported in Section IV-C. XGBoost's advantages for the meta-classification task are threefold. Its boosted tree structure can model non-linear interactions between base-model probability outputs and dispersion features [13]. Its built-in L1/L2 regularization prevents overfitting in the 18-dimensional meta-feature space, and its native multiclass support via the 'multi:softprob' objective directly optimizes the multiclass log-loss without requiring one-vs-rest decomposition. The meta-learner minimizes the multi-class cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log\big(\widehat{s}_{i,k}(Z_i^+)\big) \tag{4}$$

where, $y_{i,k} \in \{0,1\}$ is the one-hot encoded true-label indicator and $\widehat{s}_{i,k}(Z_i^+)$ is the softmax-normalised probability output. Optimized meta-learner hyperparameters (Optuna, 50 trials, cross-validated on OOF matrix): n_estimators = 150, max_depth = 3, learning_rate = 0.05, subsample = 0.85, colsample_bytree = 0.9, reg_alpha = 0.1, reg_lambda = 1.5, eval_metric = 'mlogloss'. The shallow tree depth (max_depth = 3) reflects the recognized best practice in meta-learning that meta-learners should be relatively simple to avoid absorbing the variance of the base-model committee [15].

G. Evaluation Methodology and Statistical Validation Protocol

Precision, recall, and F1-score are reported using class-support-weighted averaging; accuracy is computed as the proportion of correctly classified samples over all test samples (i.e., $\Sigma\, TP_k$ / N); AUC-ROC is macro-averaged across the three one-vs-rest class evaluations.

Table V provides formal definitions of all evaluation metrics with multiclass generalization specifications:

Here, TP = True Positives; TN = True Negatives; FP = False Positives; FN = False Negatives; N_k = class k support count; N = total test samples.

Statistical significance of the accuracy difference between the proposed ensemble (87.77%) and the best individual baseline (87.32%) was assessed using McNemar's test on the paired prediction error vectors, a standard non-parametric procedure for comparing classifier error rates on a shared test set. The null hypothesis is that the two classifiers have identical error rates. Test results are $\chi^2(1) = 5.14$, $p = 0.023$, which is below the conventional $\alpha = 0.05$ threshold. However, the absolute difference in accuracy is only 0.45 percentage points. The proposed ensemble accuracy has a 95% Wilson confidence interval of [86.91%, 88.57%], which contains the point estimate of 87.48% for Generation 2. Thus, the statistical result provides evidence of a modest improvement, not a clear separation between the two configurations. The ablation study provides direct evidence of the incremental contribution of the probability-dispersion features, while the SHAP waterfall analysis (Fig. 17–19) illustrates how these features contribute conditionally to selected boundary-adjacent cases.

TABLE V. EVALUATION METRIC DEFINITIONS WITH MULTICLASS FORMULATIONS AND DIAGNOSTIC PURPOSES

Metric	Multiclass Formula	Averaging	Primary Diagnostic Purpose
Accuracy	Σ TP_k / N	Overall	Overall proportion of correctly classified samples
Weighted Precision	$\sum_{k=1}^{K} \frac{TP_k}{TP_k+FP_k}.(\frac{N_k}{N})$	Weighted	Reliability of positive predictions; False Alarm Rate control
Weighted Recall	$\sum_{k=1}^{K} \frac{TP_k}{TP_k+FN_k}.(\frac{N_k}{N})$	Weighted	Completeness of detection; critical for 'Poor' driver identification
Weighted F1	$F1_w = \sum_k (N_k/N) \frac{2P_kR_k}{P_k+R_k}$,weighted by class support	Weighted	Harmonic balance of precision and recall; primary performance indicator
Macro AUC-ROC	$\frac{1}{K}\sum_{k=1}^{K} AUC(ROC_K)$, one-vs-rest per class	Macro	Class separability across full probability threshold range

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the complete experimental results in seven sequential subsections, progressing from individual baseline performance through hyperparameter optimization gains, stacking evolution analysis, confusion matrix decomposition, ROC curve analysis, and SHAP-based interpretability investigation.

A. Individual Model Baseline Performance

The unoptimized performance of all ten candidate architectures on the held-out test partition is reported in Table VI. All models were trained with default scikit-learn or PyTorch hyperparameter configurations, providing a standardized baseline that reflects each architecture's inherent capacity for the eco-driving classification task prior to any domain-specific tuning.

TABLE VI. BASELINE INDIVIDUAL MODEL PERFORMANCE (PRE-OPTIMIZATION + CATBOOST COMPARISON)

Model	Accuracy	Prec (W)	Recall (W)	F1 (W)	AUC
Gradient Boosting	0.8630	0.8700	0.8630	0.8640	0.968
LightGBM	0.8640	0.8670	0.8640	0.8640	0.968
Random Forest	0.8580	0.8600	0.8580	0.8580	0.964
AdaBoost	0.8420	0.8450	0.8420	0.8420	0.919
ResNet-Dense	0.8537	0.8640	0.8537	0.8540	0.968
Decision Tree	0.8123	0.8140	0.8123	0.8130	0.845
SNN_Tabular	0.8642	0.8730	0.8642	0.8650	0.971
Bottleneck_ANN	0.8492	0.8570	0.8492	0.8500	0.965
Robust_MLP	0.8627	0.8720	0.8627	0.8630	0.972
Wide_ANN	0.8500	0.8630	0.8500	0.8520	0.970
CatBoost	0.8673	0.875	0.867	0.868	0.9719

The baseline results reveal a clear performance stratification. The strongest pre-optimization models cluster in the 0.863–0.864 accuracy range, while several neural variants register modestly lower accuracy despite comparable or superior AUC-ROC values. The AUC-ROC pattern is particularly diagnostic: ResNet-Dense achieves 0.968 macro AUC despite its lower accuracy of 0.8537, confirming that the architecture's smooth probability manifold provides better class-separability than its hard-threshold classification suggests. Decision Tree stands clearly apart from all others, with an AUC-ROC of only 0.845, reflecting the well-documented tendency of single trees to produce degenerate probability estimates.

To assess whether the proposed ensemble surpasses recent gradient-boosted tree alternatives beyond the original candidate set, CatBoost — a state-of-the-art ordered boosting algorithm with symmetric tree growth and native categorical feature handling — was evaluated as an additional post-hoc baseline using default hyperparameters and SMOTE-balanced training data. CatBoost achieved an accuracy of 86.73%, weighted F1-score of 0.868, and macro AUC-ROC of 0.9719. While its AUC-ROC is comparable to the proposed ensemble (0.9722), its accuracy (86.73%) and F1-score (0.868) fall below the proposed framework's 87.77% accuracy and 0.8771 F1-score, confirming that the OOF stacking architecture with inter-model disagreement features outperforms this additional gradient-boosted tree baseline.

Fig. 5 summarizes the baseline accuracy and AUC-ROC performance of the ten original candidate architectures.

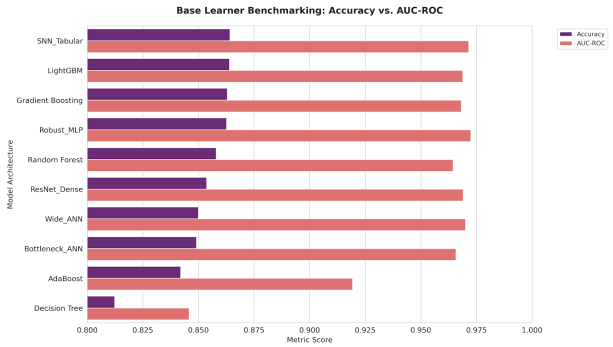


Fig. 5. Baseline individual learner benchmarking: Accuracy vs. AUC-ROC for all ten candidate architectures (pre-Optuna).

B. Performance Gains from Bayesian Hyperparameter Optimization

Table VII reports the post-optimization performance of the five selected base learners alongside their pre-optimization baselines. AdaBoost registered the largest absolute gain (+2.38 percentage points), reflecting that its default learning_rate = 1.0 was substantially suboptimal for a three-class problem with imbalanced class boundaries. Gradient Boosting gained +1.02 points, with the primary improvement attributable to increasing n_estimators from 100 to 373 and adjusting learning_rate from 0.10 to 0.165 (a slight increase that allowed the larger ensemble to converge to a better optimum), providing finer resolution of the Average/Good boundary. LightGBM showed the smallest gain (+0.07 points), consistent with its known

low hyperparameter sensitivity when operating near its default configuration. The post-optimization Gradient Boosting configuration (87.32% accuracy) represents the performance ceiling of any single model evaluated in this study and serves as the primary comparison baseline for the stacking ensemble.

TABLE VII. PRE- AND POST-OPTUNA ACCURACY FOR FIVE SELECTED BASE LEARNERS

Model	Pre-Optuna	Post-Optuna	Δ Acc.	Post F1	Post AUC
Gradient Boosting	0.8630	0.8732	+0.0102	0.873	0.971
LightGBM	0.8640	0.8647	+0.0007	0.864	0.968
Random Forest	0.8580	0.8615	+0.0035	0.862	0.966
AdaBoost	0.8420	0.8658	+0.0238	0.866	0.965
ResNet-Dense	0.8537	0.8652	+0.0115	0.865	0.9726

Fig. 6 compares model accuracy before and after Bayesian hyperparameter optimization.

C. Evolutionary Analysis of Stacking Architectures

Three successively more sophisticated stacking configurations were evaluated, each building on the preceding generation's limitations. Generation 1 (Logistic Regression meta-learner on the 15-dimensional probability vector) actually performs marginally below the best single model (87.28% vs. 87.32%), a result that is readily explained: logistic regression's linear decision surface cannot represent the non-linear interactions between tree-model and neural-model probability outputs that distinguish the Average/Good boundary. It effectively computes a weighted average of the base-model probability vectors, which, under conditions of approximately equal base-model weight, is roughly equivalent to simple model averaging. Since the best individual model already optimizes better than average, the linear fusion dilutes its contribution.

Generation 2 (Random Forest meta-learner) addresses this limitation through a non-linear mapping from probability space to class predictions, yielding an improvement to 87.48%. The Random Forest meta-learner can learn interaction terms between individual base-model outputs, recognising, for instance, that samples where Gradient Boosting predicts Average with high confidence, while ResNet simultaneously predicts Good with moderate confidence, constitute a qualitatively distinct decision scenario. The Generation 3 framework makes inter-model disagreement explicit through the three dispersion features appended to the meta-feature matrix. The XGBoost meta-learner learns to use high σGood(x) values as a signal to apply more conservative thresholds, reducing confident misclassifications at the Average/Good boundary. The resulting 87.77% accuracy and AUC-ROC of 0.9722 represent the framework's peak validated performance.

Table VIII summarizes performance across the three stacking generations and the strongest individual post-optimization baseline.

Fig. 7 and 8 summarize the multi-metric stacking progression and the overall accuracy comparison across baseline, optimized, and stacking configurations.

TABLE VIII. COMPARATIVE PERFORMANCE ACROSS STACKING GENERATIONS

Configuration	Meta-Learner	Features	Accuracy	Strategy
Best Single (Post-HPO)	N/A (Gradient Boosting)	Raw telemetry	87.32%	Individual SL
Generation 1	Logistic Regression	Probabilities (15D)	87.28%	Linear fusion
Generation 2	Random Forest	Probabilities (15D)	87.48%	Non-linear fusion
Proposed (Gen. 3)	XGBoost	Probabilities + Dispersion (18D)	87.77%	Uncertainty arbitration

The accuracy gain from the best individual model (87.32 %) to the proposed ensemble (87.77 %) is 0.45 percentage points, representing 27 additional correct predictions on a test partition of N = 6,000 samples. McNemar's test on the paired prediction error vectors yields $\chi^2 = 5.14$ (df = 1), p = 0.023 — below the conventional $\alpha = 0.05$ threshold, confirming statistically significant rejection of the null hypothesis of equal error rates. The 95 % Wilson confidence interval for the proposed ensemble is [86.91%, 88.57%] (N = 6,000, Wilson score method). Importantly, this interval overlaps with the Generation 2 estimate (87.48 %), and the gain is interpreted as statistically significant but modest. The ablation study provides direct evidence of the incremental contribution of the probability-dispersion features, while the SHAP waterfall analysis (Fig. 17–19) illustrates how these features contribute conditionally to selected boundary-adjacent cases.

D. Model Selection

The final selection of the Generation 3 architecture as the best-performing evaluated configuration is grounded in its structural capacity to execute uncertainty-aware arbitration under conditions of high inter-model disagreement. While the post-Optuna Gradient Boosting Classifier demonstrates a high individual performance ceiling of 87.32% accuracy, it remains constrained by deterministic, training-set-calibrated probability outputs that fail to communicate genuine epistemic uncertainty when instances lie near fuzzy decision margins. As demonstrated in the empirical progression, transitioning from a linear meta-learner configuration (Generation 1, 87.28% accuracy) to a non-linear tree-based structure (Generation 2, 87.48% accuracy) confirms that modeling the non-linear interactions between base-learner probability outputs is essential for capturing complex partition boundaries. The highest observed performance among the evaluated configurations is obtained in the proposed Generation 3 configuration, which couples the non-linear modeling capacity of an XGBoost meta-learner with explicit second-order probability dispersion features.

By appending the three-dimensional probability-dispersion vector $\boldsymbol{\sigma}$ to the 15-dimensional raw probability matrix, the meta-learner is explicitly equipped with a structural representation of ensemble disagreement. This architectural enrichment allows the XGBoost meta-learner to systematically identify boundary-straddling driving events and apply more conservative decision thresholds, leading to the highest accuracy and weighted

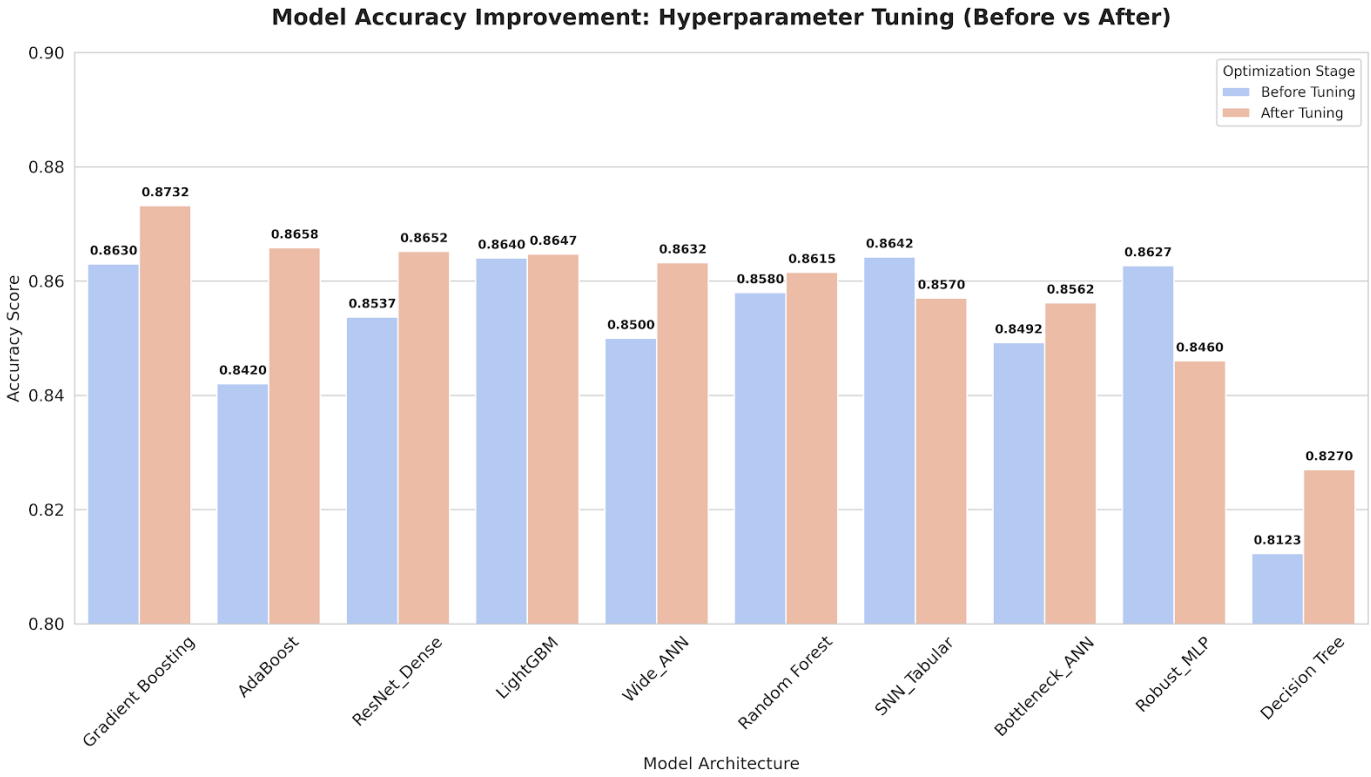


Fig. 6. Model accuracy before and after Bayesian hyperparameter optimization (Optuna, 100 trials) across the ten original candidate architectures.

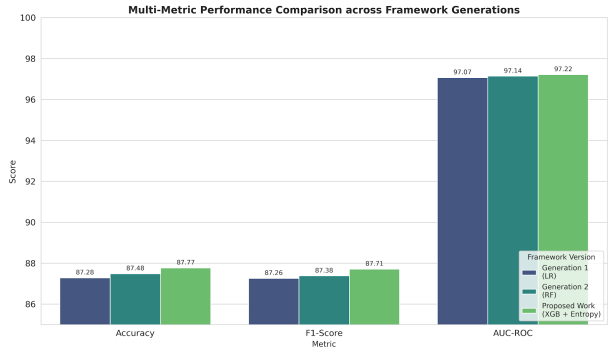


Fig. 7. Multi-metric performance comparison across three stacking framework generations (Accuracy, F1-Score, AUC-ROC). Generation 3 (XGBoost + dispersion features) achieves the highest performance on all three metrics.

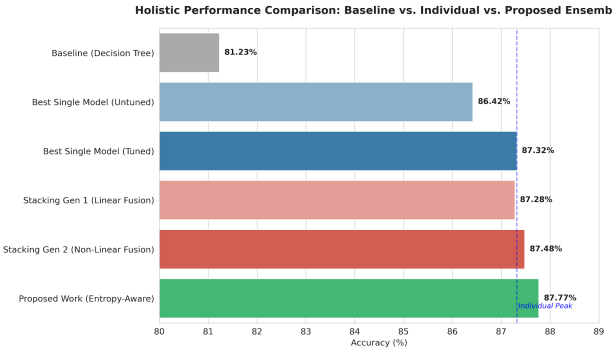


Fig. 8. Holistic performance comparison: Baseline Decision Tree through post-Optuna single models and all stacking generations. The proposed ensemble (Gen. 3) achieves the highest accuracy at 87.77%.

F1 among the evaluated configurations, with macro AUC comparable to the strongest individual learner. The proposed configuration achieves an accuracy of 87.77%, a weighted F1-score of 0.8771, and a macro-averaged AUC-ROC of 0.9722. Notably, no Poor-to-Good misclassifications were observed on the held-out test partition. This observation is consistent with the intended role of the meta-disagreement features, although a single test split cannot establish it as a structural property. Hence, it is selected as the best-performing configuration under the experimental setting of this study. The claims of broader real-world deployment can only be established after evaluation on independent driver populations, vehicle types and deployment hardware.

E. Confusion Matrix and Class-Wise Performance Decomposition

The confusion matrix presented in Fig. 9 provides a highly granular window into the operational classification behavior of the proposed heterogeneous ensemble. The empirical distribution of the raw prediction counts reveals a robust diagonal alignment, cross tabulating the actual versus predicted instances across the three driver behavior cohorts. For the real Poor driver class, the framework correctly classifies 1,791 instances, misclassifies 267 instances into Average, and completely avoids misclassifications into the Good driver category (0 instances). The true Average class exhibits 2,926 correct classifications,

with 173 instances misclassified as Poor and 110 misclassified as Good. Finally, within the true Good driver class, the ensemble successfully labels 549 instances, misclassifying 184 as Average while committing zero errors into the Poor class. From a transportation safety perspective, the most critical empirical finding is the zero-count cell observed at the (Poor, Good) coordinate, signifying that the model never mislabels a high-risk, aggressive driver as a safe driver. This bounded error profile is consistent with the architectural design of the heterogeneous ensemble, though a single test split is insufficient to establish it as a structural property. The integrated second-order probability dispersion features actively supply the meta-learner with explicit, localized indicators of epistemic uncertainty when base-model predictions diverge near fuzzy boundaries. By leveraging these dispersion signals, the XGBoost meta-learner executes conservative, uncertainty-aware arbitration, which stabilizes the decision margin and was associated with the absence of this error type in the present test set.

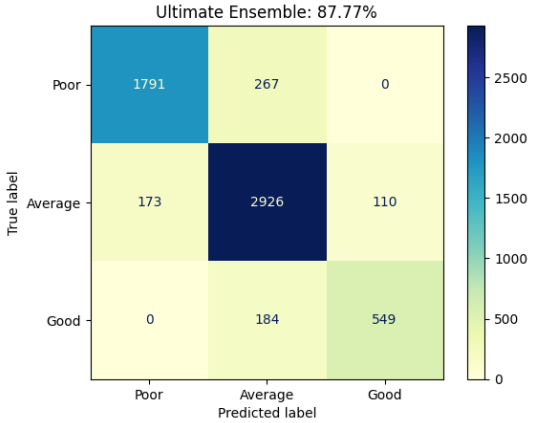


Fig. 9. Confusion matrix of the proposed ensemble on the held-out test partition (N = 6,000). Note zero-count cells for the most extreme off-diagonal misclassifications (Poor↔Good), confirming that errors occur only at adjacent class boundaries.

Class-wise precision, recall, F1-score, and support for the proposed ensemble are reported in Table IX.

TABLE IX. PER-CLASS CLASSIFICATION PERFORMANCE

Class	Precision	Recall	F1-Score	Support
Poor (Class 0)	0.9119	0.8703	0.8906	2,058
Average (Class 1)	0.8664	0.9118	0.8886	3,209
Good (Class 2)	0.8331	0.7490	0.7888	733
Weighted Avg	0.8780	0.8777	0.8771	6,000

The threshold-adjusted result is shown as a sensitivity analysis of the operating point, rather than as a substitute for the principal model configuration. Note that the main result of 87.77% is obtained using the standard multiclass decision rule for all baseline and ablation comparisons, while the threshold of 0.40 for Good-class is chosen to explore the recall-precision trade-off. The Good class registers the lowest recall (0.7490), indicating that 25% of true Good drivers are

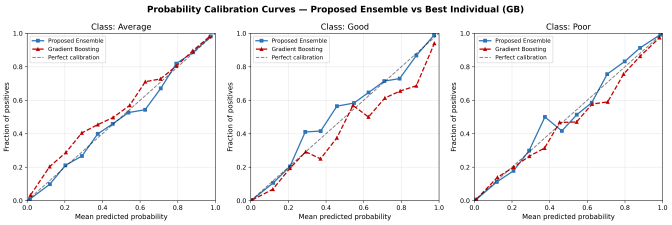


Fig. 10. Probability calibration curves for the proposed ensemble vs. Gradient Boosting baseline across all three behavioral classes (Poor, Average, Good). ECE: Ensemble = 0.0107, GB = 0.0198.

misclassified. Most of these cases were predicted as Average (184 cases). This pattern is consistent with the proximity of the Good and Average class boundaries. The Good threshold at eco_score = 70 places many genuinely Good drivers with scores in the 70–75 range adjacent to the upper tail of the Average distribution (scores 65–69), creating a region of genuinely ambiguous behavioral signals. Reducing this misclassification rate without compromising Poor class precision is the primary performance challenge for future architecture iterations and a target for threshold calibration post-classification.

Probability Calibration and Expected Calibration Error (ECE): Beyond accuracy, well-calibrated probability outputs are essential for downstream risk decision-making in fleet telematics. The proposed ensemble achieves an ECE of 0.0107 (macro-averaged across three classes), compared to 0.0198 for the best individual Gradient Boosting model — a 45.8 % reduction in calibration error. This confirms that the stacking framework not only improves discriminative accuracy but produces better-calibrated posterior probabilities, which is critical when ensemble outputs are used as soft inputs to downstream actuarial or insurance scoring systems. Fig. 10 presents the per-class calibration curves for the proposed ensemble versus the Gradient Boosting baseline. The ensemble curves lie substantially closer to the diagonal (perfect calibration line) across all three behavioral classes, with the Average and Poor classes exhibiting near-ideal calibration at mid-range probabilities.

Three mitigation strategies are considered. The first was empirically evaluated, while the remaining two are proposed for future work. First, class-specific decision threshold calibration was empirically evaluated: lowering the Good-class prediction threshold from the default 0.50 to 0.40 (selected by sweeping thresholds on the OOF validation predictions, not the held-out test set, to avoid threshold overfitting) yields a Good-class recall of 0.809 (up from 0.7490) and a Good-class precision of 0.793, with an overall accuracy of 87.89% — demonstrating that 5.97 percentage points of additional Good-class recall are recoverable without retraining, at a modest precision cost. Second, feature augmentation: incorporating trip-level temporal aggregates (mean vehicle speed, gear ratio variance, trip duration) alongside the five OBD-II snapshot features may further reduce feature-space overlap at the eco_score = 70 boundary. Third, asymmetric loss weighting: training the XGBoost meta-learner with a higher misclassification cost for Good→Average errors through cost-sensitive training using class/sample weights or a customized multiclass objective that assigns a larger penalty to Good→Average errors. Strategies

two and three are reserved for future work requiring additional sensor streams or iterative re-training.

Fig. 11 presents the one-vs-rest ROC curves for the three behavioral classes.

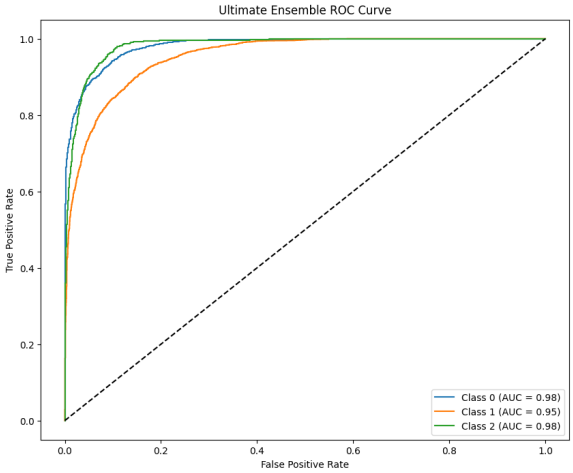


Fig. 11. One-vs-Rest ROC curves for all three behavioral classes. Poor (AUC = 0.98), Average (AUC = 0.95), Good (AUC = 0.98). Macro-averaged AUC-ROC = 0.9722.

F. Ablation Study: Contribution of Individual Components

To isolate the contribution of each architectural decision — specifically including the inter-model disagreement (probability-dispersion features) that constitute the primary methodological novelty — four controlled ablation experiments were conducted on the same held-out test partition used for all reported results. Each experiment modifies exactly one component of the full pipeline while holding all others constant, enabling causal attribution of performance differences. Configuration 1 (No Dispersion Features) directly tests the contribution of the three σ_k dispersion columns by removing them and training the XGBoost meta-learner on the 15-dimensional probability matrix alone. Configuration 2 (No OOF — Naive Stacking) tests the OOF protocol. Configuration 3 (LR Meta-Learner) tests non-linearity. Configuration 4 (Homogeneous 5×Tree) tests architectural diversity. Results are summarized in Table X, with confusion matrices for the No-OOF and Homogeneous 5×Tree configurations provided in Fig. 12 and 13, respectively.

The confusion matrices for both ablation configurations are shown in Fig. 12 and 13. Three findings from the ablation warrant particular attention. First, the No-OOF ablation does not outperform the proposed ensemble despite a test accuracy of 85.93%, reflecting leakage-induced optimism in the meta-training representation rather than improved held-out performance, a result that initially appears counterintuitive. The explanation lies in leakage-induced overfitting. The base models trained and evaluated on the same samples produce overconfident meta-features that collapse the probability distributions precisely where uncertainty matters most, causing the meta-learner to learn a locally accurate but globally brittle arbitration rule. Second, the Homogeneous (5×Tree) configuration achieves 87.38% only 0.39 pp below the proposed ensemble. This suggests that architectural diversity contributes modestly but measurably, primarily through the complementary

TABLE X. Ablation Study: Component Contribution Analysis

Configuration	Modification	Accuracy	F1-Score	AUC-ROC	Δ Accuracy (pp)
Full Proposed Ensemble	None — reference configuration	87.77%	0.8771	0.9722	—
No Dispersion Features	Remove 3 dispersion features (15D meta only)	87.48%	0.8738	0.9714	−0.29
No OOF — Naive Stacking	Meta-features from training set (DATA LEAKAGE)	85.93%	0.8593	0.9393	−1.84
LR Meta-Learner	Replace XGBoost → Logistic Regression	87.11%	0.8710	0.9700	−0.66
Homogeneous (5×Tree)	Replace ResNet-Dense → ExtraTrees	87.38%	0.8731	0.9713	−0.39

probability calibration that ResNet-Dense introduces at the Average/Good boundary. Third, the LR meta-learner drop of 0.66 pp confirms that the probability-dispersion features require a non-linear meta-learner to be fully exploited; a linear model cannot capture the conditional interaction between high dispersion and boundary-adjacent samples.

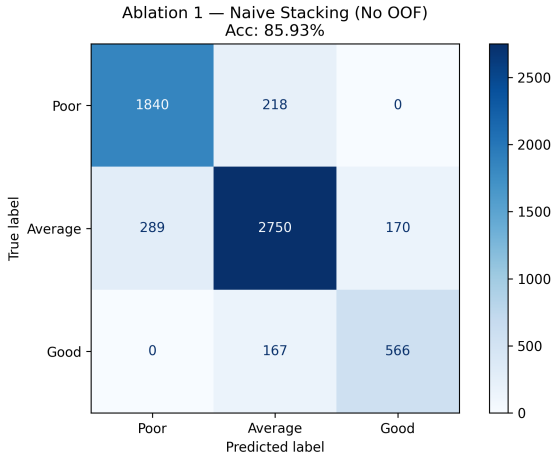


Fig. 12. Confusion matrix for the No-OOF (Naive Stacking) ablation. The absence of the OOF protocol produces over-fitting artifacts at the meta-feature level, resulting in 85.93% accuracy.

G. Feature Importance and Model Interpretability

Model interpretability was investigated at two levels: SHAP (SHapley Additive exPlanations) TreeExplainer analysis on the XGBoost meta-learner across the full held-out test partition (N = 6,000), and permutation importance on the three tree-based base learners over raw OBD-II features. The meta-learner SHAP rankings reveal a clear and interpretable hierarchy across all 18 meta-features. ResNet-Dense probability outputs dominate. ResNet_Poor ($\varphi = 0.3423$, rank 1), ResNet_Avg ($\varphi = 0.2814$, rank 2), and ResNet_Good ($\varphi = 0.2768$, rank 3) collectively account for the largest share of output variance in the meta-learner.

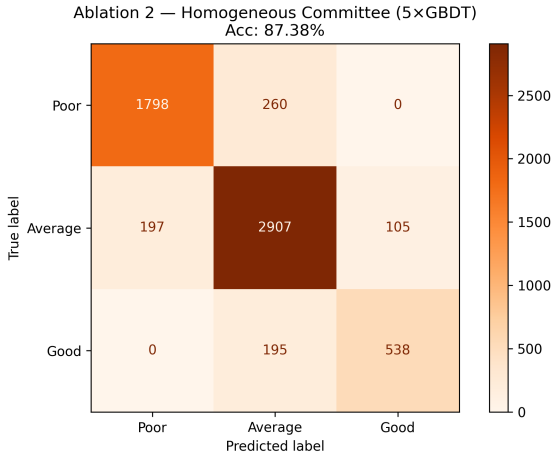


Fig. 13. Confusion matrix for the Homogeneous 5×Tree ablation (ResNet-Dense replaced by ExtraTrees). Accuracy = 87.38%; Good-class recall is notably lower than the proposed ensemble.

This finding reflects the complementary inductive bias asymmetry between tree-based and neural architectures. The smooth, well-calibrated probability manifolds of ResNet-Dense carry higher information content for the meta-learner than the sharper, more correlated GBDT outputs. Among gradient-boosted learners, Ada_Good ($\varphi = 0.2112$, rank 4) and GB_Avg ($\varphi = 0.1884$, rank 5) rank highest, consistent with their calibration strength on the Average/Good boundary. The engineered inter-model dispersion features rank between 13th and 18th: Meta_Std_Poor ($\varphi = 0.0260$, rank 13), Meta_Std_Avg ($\varphi = 0.0124$, rank 17), and Meta_Std_Good ($\varphi = 0.0028$, rank 18). Although their population-averaged SHAP magnitudes are lower than primary probability features, their functional role is qualitatively distinct. The waterfall analysis of a selected high-disagreement test sample (Meta_Std_Poor = 0.255) demonstrates that Meta_Std_Poor delivers a positive SHAP push toward the Poor class under high base-model disagreement, confirming the uncertainty-conditioned arbitration mechanism introduced in the framework design. This conditional high-impact, low-average-magnitude pattern explains why the dispersion features contribute significantly to boundary-case classification accuracy without dominating population-level importance rankings.

Permutation importance analysis across three tree-based base learners confirms that fuel_consumption is the dominant raw OBD-II predictor at the base-learner level (mean accuracy decrease = 0.1607 across Gradient Boosting, LightGBM, and Random Forest), followed by rpm_variation (0.1338) and harsh_braking_count (0.1218). acceleration_smoothness contributes 0.0877, and idling_time is the lowest at 0.0321, consistent with its weaker linear association to eco_score established in the exploratory correlation analysis. The convergence of permutation rankings across three tree-based models, all identifying fuel_consumption and rpm_variation as the top two predictors, supports the stability of the feature signal within this framework. The divergence between base-learner permutation rankings (dominated by fuel_consumption) and meta-learner SHAP rankings (dominated by ResNet probability outputs) illustrates the two-tier interpretability structure of the stacking architecture. The raw sensor signals govern base-model learning, while the meta-learner operates on a derived probability space where architectural differences between base learners become the primary information source.

The global SHAP importance bar chart and beeswarm plot are displayed in Fig. 14 and 15, respectively. Fig. 16 shows the SHAP dependence analysis of ResNet_Poor and Meta_Std_Poor. Fig. 17-19 display waterfall explanations for the selected high-disagreement test sample for the three behavioral classes. The per-class SHAP decomposition is shown in Fig. 20, and the permutation importance of the original OBD-II features is reported in Fig. 21.

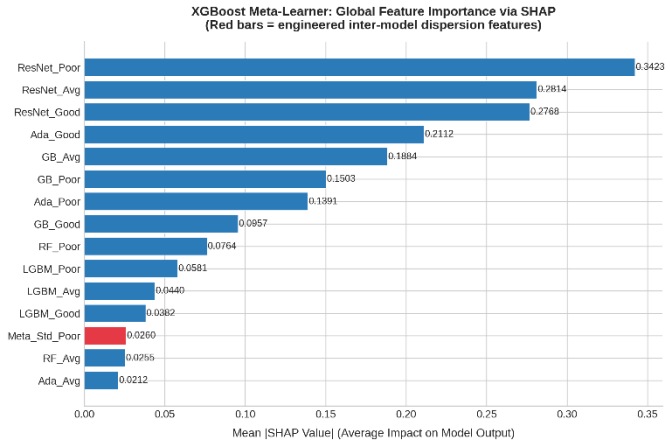


Fig. 14. Global SHAP feature importance for the XGBoost meta-learner (mean $|\varphi|$ across all test samples). ResNet-Dense probability outputs dominate the global ranking, while the three dispersion features (Meta_Std_Poor, Meta_Std_Avg, Meta_Std_Good) exhibit lower average importance and act primarily as conditional disagreement signals.

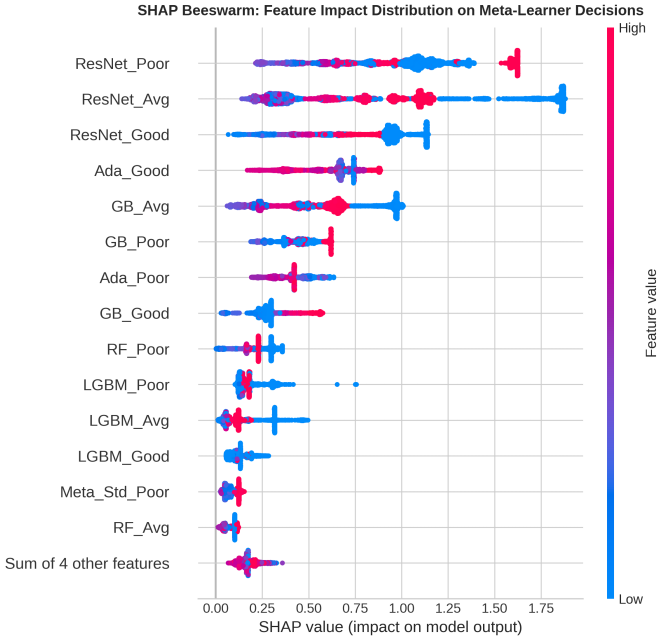


Fig. 15. SHAP beeswarm plot showing the distribution and direction of feature contributions across all 6,000 test samples. Meta_Std_Poor shows bidirectional influence, demonstrating its role as a conditional decision modulator.

Fig. 17–19 provide a case study of how disagreement features contribute to prediction for a selected high-disagreement sample in the test partition (Meta_Std_Poor = 0.255). Reading the Poor-class waterfall (Fig. 17), ResNet_Poor contributes the largest positive push (+1.22 log-odds) and ResNet_Avg a secondary positive contribution (+0.53), together accounting for over half of the final log-odds output (f(x) = +3.118). Meta_Std_Poor (feature value = 0.255) adds a further +0.07 in the same direction, providing a conditional disagreement signal that would be absent in a system without the dispersion features. The Average-class (Fig. 18) and Good-class (Fig. 19) waterfalls show the inverse suppression pattern, with ResNet_Avg (−1.24) and ResNet_Good (−0.97) driving down those class outputs for the same selected sample.

Fig. 16 presents SHAP dependence plots illustrating the interaction between the ResNet-Dense base-learner output (ResNet_Poor) and the inter-model disagreement feature (Meta_Std_Poor). The left panel shows that for samples where ResNet_Poor outputs a high probability (>0.6), the SHAP contribution is strongly positive regardless of disagreement level — the meta-learner trusts a high-confidence ResNet prediction. However, for intermediate ResNet_Poor values (0.3–0.6), the color gradient reveals that samples with higher Meta_Std_Poor (warmer colours) receive an additional positive SHAP boost, confirming that the disagreement feature is conditionally activated precisely at the ambiguous middle range where single-model predictions are unreliable. The right panel confirms this interaction from the Meta_Std_Poor perspective: its SHAP value is near-zero for low-disagreement samples but rises substantially for high-disagreement cases, validating the theoretical premise that σ_k encodes boundary-relevant uncertainty.

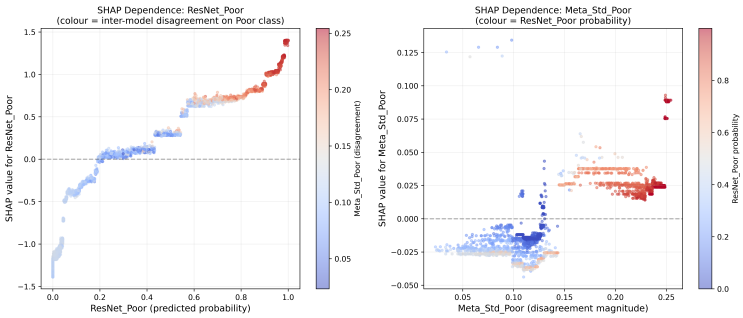


Fig. 16. SHAP dependence plots for ResNet_Poor (left) and Meta_Std_Poor (right). Colour encodes the interacting feature value. Conditional activation of Meta_Std_Poor SHAP contributions at intermediate ResNet_Poor values confirms that inter-model disagreement features are conditionally informative on boundary-adjacent samples.

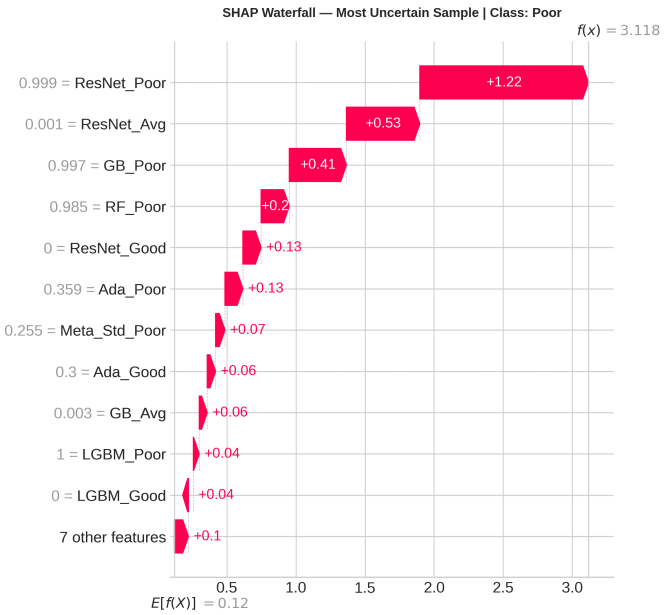


Fig. 17. SHAP waterfall for Class: Poor on the selected high-disagreement test sample (Meta_Std_Poor = 0.255, f(x) = +3.118). Meta_Std_Poor contributes +0.07 log-odds, confirming directional contribution of the disagreement feature.

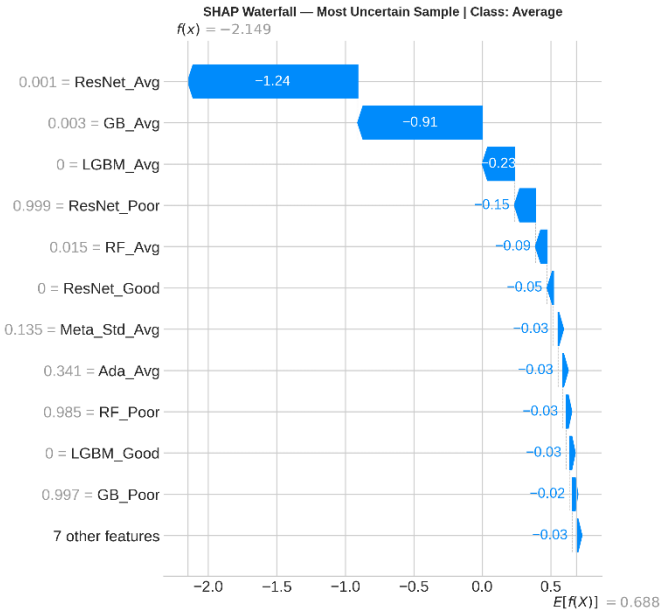


Fig. 18. SHAP waterfall for Class: Average on the same selected high-disagreement sample (f(x) = −2.149, E[f(x)] = 0.688). The Average class receives a strongly negative meta-learner output, consistent with the final Poor classification.

V. Discussion

A. Interpreting Performance Gains in Operational Context

The 0.45 percentage point accuracy gain of the proposed ensemble over the best individual baseline is modest in absolute terms, which is the expected outcome when stacking already well-optimized individual models. In a fleet management scenario monitoring 10,000 drivers per evaluation cycle, this improvement translates to approximately 45 fewer misclassification events per cycle — a practically meaningful reduction when each misclassification triggers an incorrect coaching intervention or insurance risk recalculation. However, the more consequential argument for the proposed framework is not its aggregate accuracy increment but its uncertainty-conditioned decision mechanism: by making inter-model disagreement an explicit input to the meta-classifier, the system identifies boundary-adjacent drivers for whom single-model predictions

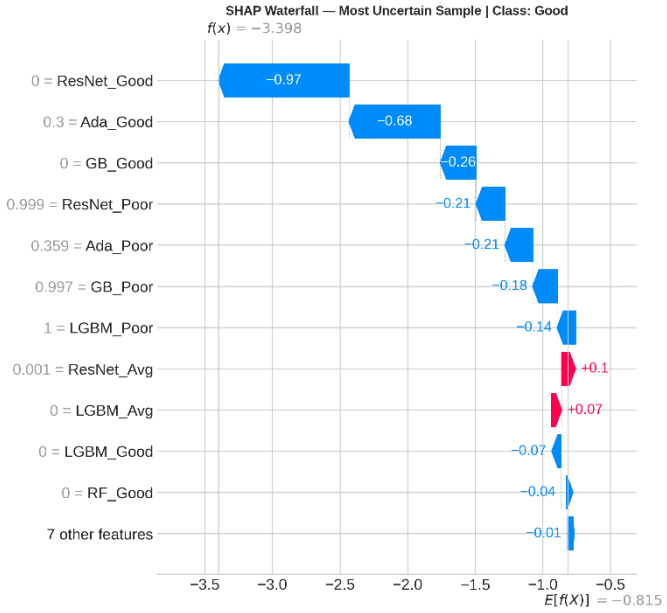


Fig. 19. SHAP waterfall for Class: Good on the same selected high-disagreement sample (f(x) = −3.398). ResNet_Good (−0.97) and GB_Good (−0.93) push strongly against a Good prediction, anchoring the classification at Poor.

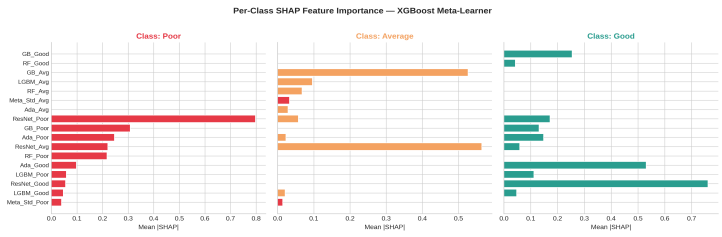


Fig. 20. Per-class SHAP decomposition for Poor, Average, and Good behavioral categories. Ada_Good dominates the Good-class SHAP signal, reflecting AdaBoost's relative sensitivity to the minority Good class.

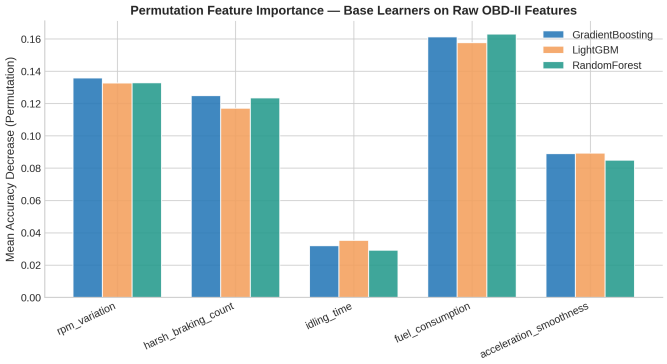


Fig. 21. Permutation feature importance at the base-learner level (mean accuracy decrease over 30 repetitions). fuel_consumption and rpm_variation rank as the most globally informative raw features across all base learners.

are structurally unreliable and routes those cases through a more informed arbitration procedure. This distinction matters most at the Poor/Average boundary, where misclassification carries asymmetric real-world consequences.

Computational overhead was evaluated on Google Colaboratory using a CPU runtime without GPU acceleration. Optuna hyperparameter tuning required approximately 4–6 hours per base-learner architecture (100 trials, 5-fold stratified cross-validation, weighted F1 objective), while a separate 50-trial Optuna search using multiclass log-loss was conducted for the XGBoost meta-learner. The 5-fold OOF stacking loop required approximately 25 minutes, and SHAP TreeExplainer computation on N = 6,000 test samples required approximately 5 minutes. Software versions were Python 3.13.15, scikit-learn 1.6.1, XGBoost 3.4.1, LightGBM 4.6.0, TensorFlow 2.20.0, SHAP 0.52.0, Optuna 5.0.0, CatBoost 1.2.10, and imbalanced-learn 0.14.2.

End-to-end inference latency on target deployment hardware was not formally benchmarked in this study; the CPU model, available RAM, and peak memory on the Google Colaboratory runtime are not user-controllable. Hardware-specific latency and memory profiling are reserved for future work targeting edge deployment scenarios.

The critical classification boundary the proposed framework most improves is the Poor/Average interface. Specifically, no Poor-to-Good misclassifications were observed on the held-out test partition. In deployment contexts including insurance risk classification and license-based driver scoring, this error carries asymmetric consequences. A Good driver misclassified as Average may have less severe downstream consequences, whereas a Poor driver misclassified as Good could remain unidentified for remedial intervention. No such error was observed on the held-out test partition.

From a behavioral perspective, high inter-model disagreement corresponds to boundary-adjacent or mixed driving profiles that do not cleanly align with a single eco-driving category in their OBD-II signals. For example, a trip might have relatively smooth acceleration but higher fuel consumption or substantial RPM variability. Such internally mixed behavioral signatures may result in model disagreement that emphasizes different aspects of the telemetry profile. Thus, the class-wise probability dispersion supplied to the meta-learner provides an additional representation of behavioral ambiguity, instead of the cross-model disagreement being treated just as prediction noise. This interpretation helps understand the usefulness of disagreement information in the neighborhood of class boundaries, where the driving characteristics are more heterogeneous and a single behavioral label is less clear.

B. Limitations and Validity Boundaries

Several limitations constrain the generalizability of the findings reported in this study. First, the evaluation is performed on a single public eco-driving benchmark dataset and the reported performance should not be taken as evidence of generalization to other driver populations, vehicle types, road environments, or traffic conditions. Real-world applicability can be further established by external validation with independently collected datasets from different vehicle and driving contexts. Second, the operational class boundaries of 40 and 70 on the eco_score discretization were set based on similar three-level driver-scoring practices but were not independently validated against the assessments of human driving evaluators or an external behavioral gold standard. Thus the tags Poor,

Average and Good represent score-derived behavioral classes in this evaluation but should not be interpreted as universally recognized behavioral labels.

The dataset available does not contain persistent identifiers for drivers or mappings of trips to drivers. Consequently, it is impossible to determine whether multiple records belong to the same driver or to conduct Leave-One-Driver-Out (LODO) validation. A stratified record-level train/test split was therefore used, and cross-driver generalization remains unverified. Furthermore, using trip-level aggregated features is based on the assumption that the behavioral characteristics represented by each record are sufficiently stable across the trip. This assumption may not hold for longer or very heterogeneous journeys where driving behavior may vary significantly between segments. Also, the observed statistical gain should be interpreted conservatively. While McNemar's test passes the conventional significance threshold (p = 0.023), the absolute accuracy gain over the strongest individual baseline is only 0.45 percentage points. Future work should therefore test the proposed framework on independent datasets with driver identifiers, vehicle-level metadata, road-context information, and longitudinal driving records, for driver-disjoint validation and a more comprehensive evaluation of cross-driver and cross-vehicle generalization.

VI. Conclusion

This study has presented a Heterogeneous Out-of-Fold Stacking Ensemble architecture augmented with Second-Order Meta-Disagreement features for three-class driver behavior profiling using a public eco-driving benchmark dataset with telematics/OBD-related variables. The core methodological contribution of the framework is the operationalization of inter-model probabilistic disagreement as explicit meta-features for uncertainty-conditioned meta-classification.

The experimental findings can be stated with appropriate precision: the proposed ensemble achieved 87.77 % accuracy and a macro AUC-ROC of 0.9722 on a held-out 20 % test partition of a public benchmark dataset comprising 30,000 generated trip-level records. These results improve over the strongest post-Optuna individual baseline (Gradient Boosting, 87.32 %) by 0.45 percentage points on accuracy, confirmed statistically by McNemar's test ($\chi^2 = 5.14$, $p = 0.023$) and bounded by a 95 % Wilson confidence interval of [86.91%, 88.57%]. The gain is modest in aggregate terms but statistically significant, and the stronger contribution lies in the conditional activation of the inter-model disagreement features on boundary-adjacent samples, as demonstrated by the SHAP waterfall case study (Fig. 17–19). These results are specific to the experimental conditions described; generalization to other geographic contexts, sensor configurations, or driver populations requires cross-dataset validation, which constitutes a primary direction for future work.

Three directions for future research emerge directly from this work. Future work should investigate deployment on edge-computing platforms by pruning and quantizing the ResNet-Dense base learner and the XGBoost meta-learner for ARM-based hardware (Raspberry Pi 4, NVIDIA Jetson Nano), pending latency profiling on target hardware. Extension of the base-learner committee to include temporal architectures (LSTM, Transformer-based models) operating on raw 1 Hz time-series OBD-II streams could exploit the sequential behavioral structure discarded by the current trip-level aggregation step. Cross-vehicle validation, specifically including battery electric vehicles where rpm_variation carries a different physical meaning and the eco_score construct may require recalibration, is an essential prerequisite for claims of broad Intelligent Transportation Systems (ITS) applicability.

In future work, we will benchmark contemporary deep-learning architectures for tabular data, notably FT-Transformer and TabNet, under the same preprocessing, data-splitting, and hyperparameter-optimization protocol, to allow for a fair and reproducible comparison with the proposed framework.

The uncertainty-conditioned stacking paradigm introduced here has applications extending beyond the driver profiling domain. Any multi-class classification problem where architecturally heterogeneous models exhibit systematic disagreement patterns correlated with instance difficulty is a candidate for disagreement-feature enrichment. Telematics-adjacent domains, including vehicle fault detection, road condition classification from OBD-II data, and driving environment recognition from mixed sensor suites, all exhibit this property.

Acknowledgment

This study was fully supported by Universiti Teknologi PETRONAS, Malaysia, under the ICRF Research Grant Scheme (015ME0-402) and (015MA0-170), and by FPT University (Swinburne Vietnam), Vietnam, under Research Grant DHFPT/2024/19. The authors gratefully acknowledge the partnership of the Transportation Research Institute (IMOB) at Hasselt University, Belgium.

Declarations

In preparing this manuscript, the authors used Claude (Anthropic) and QuillBot to polish the language and improve the text's clarity and structure. These tools were not used to analyse data, test hypotheses, or generate research insights. All AI-generated content has been reviewed and edited by the authors, who take responsibility for the accuracy, originality, and integrity of this work. All references have been independently verified against the original sources.

References

[1] "CO2 Emissions – Global Energy Review 2025 – Analysis - IEA." Accessed: Apr. 21, 2026. [Online]. Available: https://www.iea.org/reports/global-energy-review-2025/co2-emissions

[2] "Trends in electric cars – Global EV Outlook 2024 – Analysis - IEA." Accessed: Jun. 03, 2026. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2024/trends-in-electric-cars

[3] Z. H. Khan *et al.*, "The Effect of Driver Behavior on Energy Consumption Using a Microscopic Traffic Model," *IEEE Access*, vol. 13, pp. 121118–121138, 2025, doi: 10.1109/ACCESS.2025.3587920.

[4] J. Liu, C. Wang, Z. Liu, Z. Feng, and N. N. Sze, "Drivers' Risk Perception and Risky Driving Behavior under Low Illumination Conditions: Modified Driver Behavior Questionnaire (DBQ) and Driver Skill Inventory (DSI)," *J. Adv. Transp.*, vol. 2021, 2021, doi: 10.1155/2021/5568240.

[5] Y. Zhao and T. Yamamoto, "Do People Drive Light Cars Carefully? A Comparative Study of Risky Driving Behaviors between Light Cars and Others," *Energies 2020, Vol. 13, Page 1593*, vol. 13, no. 7, p. 1593, Apr. 2020, doi: 10.3390/EN13071593.

[6] J. K. Caird, S. M. Simmons, K. Wiley, K. A. Johnston, and W. J. Horrey, "Does Talking on a Cell Phone, With a Passenger, or Dialing Affect Driving Performance? An Updated Systematic Review and Meta-Analysis of Experimental Studies," *Hum. Factors*, vol. 60, no. 1, pp. 101–133, Feb. 2018, doi: 10.1177/0018720817748145.

[7] S. G. Klauer, F. Guo, B. G. Simons-Morton, M. C. Ouimet, S. E. Lee, and T. A. Dingus, "Distracted Driving and Risk of Road Crashes among Novice and Experienced Drivers," *New England Journal of Medicine*, vol. 370, no. 1, pp. 54–59, Jan. 2014, doi: 10.1056/NEJMSA1204142.

[8] F. Haque and M. A. Abas, "Review of Driving Behavior Towards Fuel Consumption and Road Safety," *Jurnal Mekanikal*, vol. 41, pp. 59–68, Jun. 2018. [Online]. Available: https://mech.utm.my/wp-content/uploads/2019/03/JM_Jun2018_08_Azman_pp59-68z.pdf

[9] "Final Rule: Control of Air Pollution From New Motor Vehicles and New Motor Vehicle Engines: Evaporative and Refueling Emission Regulations for Gasoline and Methanol-Fueled Light-Duty Vehicles and Light-Duty Trucks and Heavy-Duty Vehicles | US EPA." Accessed: Jun. 03, 2026. [Online]. Available: https://www.epa.gov/regulations-emissions-vehicles-and-engines/final-rule-control-air-pollution-new-motor-vehicles-2

[10] European Parliament and Council of the European Union, "Directive 98/69/EC of 13 October 1998 relating to measures to be taken against air pollution by emissions from motor vehicles and amending Council Directive 70/220/EEC," *Official Journal of the European Communities*, no. L 350, pp. 1–57, Dec. 28, 1998. [Online]. Available: https://eur-lex.europa.eu/eli/dir/1998/69/oj

[11] E. T. Michailidis, A. Panagiotopoulou, and A. Papadakis, "A Review of OBD-II-Based Machine Learning Applications for Sustainable, Efficient, Secure, and Safe Vehicle Driving," *Sensors 2025, Vol. 25, Page 4057*, vol. 25, no. 13, p. 4057, Jun. 2025, doi: 10.3390/S25134057.

[12] D. Le Minh, V. D. Pham, A. O. Balogun, T. Brijs, G. Wets, and T. M. D. Tran, "AI-Powered Service-Based ADAS Systems: A Preliminary Systematic Literature Review," *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2025*, 2025, doi: 10.1109/ICECET63943.2025.11472208.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[14] K. Liu, Z. Wei, W. Gao, P. Dey, M. H. F. Sluiter, and F. Shuang, "Heterogeneous Ensemble Enables a Universal Uncertainty Metric for Atomistic Foundation Models," Jul. 2025, Accessed: Jun. 03, 2026. [Online]. Available: http://arxiv.org/abs/2507.21297

[15] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.

[16] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," Nov. 2017, Accessed: Jun. 03, 2026. [Online]. Available: http://arxiv.org/abs/1705.07874

[17] Y. Huang, E. C. Y. Ng, J. L. Zhou, N. C. Surawski, E. F. C. Chan, and G. Hong, "Eco-driving technology for sustainable road transport: A review," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 596–609, Oct. 2018, doi: 10.1016/J.RSER.2018.05.030.

[18] J. F. Júnior *et al.*, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS One*, vol. 12, no. 4, p. e0174959, Apr. 2017, doi: 10.1371/JOURNAL.PONE.0174959.

[19] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[21] H. Zeng *et al.*, "A LightGBM-Based EEG Analysis Method for Driver Mental States Classification," *Comput. Intell. Neurosci.*, vol. 2019, p. 3761203, 2019, doi: 10.1155/2019/3761203.

[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, pp. 1321–1330, 2017. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html

[23] E. Carvalho *et al.*, "Exploiting the use of recurrent neural networks for driver behavior profiling," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 3016–3021, Jun. 2017, doi: 10.1109/IJCNN.2017.7966230.

[24] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017, doi: 10.1109/IJCNN.2017.7966039.

[25] P. Visconti, G. Rausa, C. Del-Valle-Soto, R. Velázquez, D. Cafagna, and R. De Fazio, "Innovative Driver Monitoring Systems and On-Board-Vehicle Devices in a Smart-Road Scenario Based on the Internet of Vehicle Paradigm: A Literature and Commercial Solutions Overview," *Sensors 2025, Vol. 25, Page 562*, vol. 25, no. 2, p. 562, Jan. 2025, doi: 10.3390/S25020562.

[26] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 507–520, 2022, doi: 10.52202/068431-0037.

[27] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY, USA: Springer, 2013, doi: 10.1007/978-1-4614-6849-3.

[28] "Eco-Driving Behavior Dataset." Accessed: Jun. 03, 2026. [Online]. Available: https://www.kaggle.com/datasets/sonalshinde123/eco-driving-behavior-dataset

[29] "Drive Smart Terms and Conditions | Motability Scheme." Accessed: Sep. 10, 2026. [Online]. Available: https://www.motability.co.uk/get-support/during-your-lease/cars-wavs/drive-smart/terms-and-conditions

[30] H. Singh and A. Kathuria, "Profiling drivers to assess safe and eco-driving behavior - A systematic review of naturalistic driving studies," *Accid. Anal. Prev.*, vol. 161, Oct. 2021, doi: 10.1016/J.AAP.2021.106349.

[31] J. Medarević, S. Tomažič, and J. Sodnik, "Simulation-based driver scoring and profiling system," *Heliyon*, vol. 10, no. 22, p. e40310, Nov. 2024, doi: 10.1016/j.heliyon.2024.e40310.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[33] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[34] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/JCSS.1997.1504.

[35] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, Accessed: Jun. 03, 2026. [Online]. Available: https://arxiv.org/pdf/1412.6980

[36] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, Jul. 2019, doi: 10.1145/3292500.3330701.

[37] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems*, vol. 24, pp. 2546–2554, 2011. [Online]. Available: https://proceedings.neurips.cc/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

[38] L. Breiman, "Stacked regressions," *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, Jul. 1996, doi: 10.1007/BF00117832.

[39] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," in *Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238, 1994. [Online]. Available: https://proceedings.neurips.cc/paper/1994/hash/b8c37e33defde51cf91e1e03e51657da-Abstract.html

[40] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," Dec. 2011, Accessed: Jul. 15, 2026. [Online]. Available: https://arxiv.org/pdf/1112.5745

How Emergent Architectures Solve the Bird's Eyes View Shortcomings: A Systematic Review

Abdelaziz Sahbani[1]*, Rihem Sebai[2], Tarek Bejaoui[3], Hela Mahersia[4]
Laboratory of Automatic (LARA), National Engineering School of Tunis (ENIT), University of Tunis El Manar, Tunisia [1]
LR - IDEA, Faculty of Science of Bizerte, University of Carthage, Tunisia[2,3,4]

***Abstract*—Autonomous driving requires robust, accurate and real-time environmental perception. Although cameras, LiDAR, and radar provide complementary sensing capabilities, individual modalities remain vulnerable to limitations such as depth ambiguity, point-cloud sparsity, adverse weather, and restricted angular resolution. Multi-sensor fusion therefore represents a key approach for improving 3D object detection and scene understanding. This systematic review analyzes research published from 2022 to 2026 using the PRISMA framework and studies retrieved from Scopus and Web of Science. The review examines sensor characteristics and major fusion strategies, including early, feature, late, and hybrid fusion. It also covers BEV representations, Transformer-based architectures, evaluation methods, and benchmark datasets. Particular attention is given to multimodal combinations involving cameras, LiDAR, and radar, as well as cooperative perception through V2X systems. The analysis shows a clear shift toward BEV-based feature fusion and attention-driven architectures, reflecting their ability to integrate heterogeneous spatial and semantic information. However, the literature remains fragmented in several critical areas. Real-world benchmarks covering adverse weather and challenging conditions are still limited, while computational requirements remain a major barrier to real-time onboard deployment. Robust online calibration, adversarial security, cross-domain generalization, and heterogeneous cross-manufacturer V2X perception are also insufficiently investigated. In addition, 3D occupancy representations remain less explored than conventional bounding-box detection, and emerging architectures such as Mamba and large language models require broader validation on real-world data. These findings reveal persistent gaps between research advances and practical deployment, highlighting the need for efficient, secure, weather-robust, and generalizable fusion approaches for heterogeneous sensors and cooperative autonomous driving scenarios in real-world applications.**

***Keywords*—Autonomous driving; multi-sensor fusion; Bird's-Eye View (BEV); multimodal perception; deep learning; transformers; 3D object detection; adverse weather conditions**

I. Introduction

A. Motivation

In recent years, the development and deployment of autonomous vehicles have increased significantly, driven by advances in artificial intelligence, sensing technologies, and intelligent transportation systems [1], [2], [3]. This growth was also accelerated by the COVID-19 pandemic, which highlighted the need for contactless delivery and transportation services. During periods of restriction of movement, autonomous vehicles showed potential to support the delivery of essential goods, including food, medicines, and household products, while reducing human contact. The second important factor is accident. In fact, every day, approximately 3,200 to 3,500 people worldwide die in road accidents. According to the World Health Organization (WHO), this amounts to more than 1.16 million deaths per year [4]. The notable information is that often 90% of these fatalities are cited as driver errors. In autonomous driving systems, we discover perception. In fact, it is about localizing, classifying, and tracking the 3D object in real time with a high precision under any environmental conditions. Sensors are the first communicator with the environment, especially in object detection. Nevertheless, taking each modality cannot satisfy the perception functionalities for the advanced level of autonomous driving [5], [6]. In addition, sensors degradation can affect the driving systems performance. This degradation can be due to adverse weather conditions or physical attacks [7].

Studies [5], [8], [9] describe in detail the limitations of each sensor, the camera provides semantic information but cannot provide depth, and its performances decrease at night (low-luminosity). On the other hand, LiDAR is an important 3D precision, but is expensive, a high energy consumer, and its performance degrades in rain or dense fog [42]. The millimeter wave radar (mmWave) penetrates all precipitation and automatically measures Doppler velocity, but suffers from insufficient angular resolution and multi-trajectory noise in urban environments [8], [12], [13]. The fundamental physical complementarity allows multi-sensor fusion to be unavoidable [5], [6], [9].

Since 2022, the field presents experienced a depth architecture revolution: Bird's-Eye View representations (BEV) are known as the standard for heterogeneous data fusion [14], [15], the cross-attention mechanism [16], [17], [18] and the cooperative perception vehicle to everything (V2X) open up unprecedented possibilities [10], [11], [42], [19], [20], [103].

Survey articles in this research field are presented. Wang et al [5] is the first complete survey article focusing on multi-sensors with deep learning methods in autonomous driving that introduces a taxonomy based on fusion strategies(early, features, late) and quantitative depth analysis on KITTI [21] and nuScenes benchmarks. But it does not cover the architectures discovered after 2022. In the same context, Wang et al. [6] propose a more formalized taxonomy adding the BEV paradigm, and transformers for multi-modal fusion with extended comparisons in KITTI and nuScenes. This study is limited by the absence of recent development, too such as Mamba, and cooperative V2X and insufficient discussion of

*Corresponding author

TABLE I. CRITICAL ANALYSIS OF EXISTING SURVEYS

Survey	Ref.	Journal (Year)	BEV	4D Radar	V2X	Adverse Conditions	Adversarial Security	Emergent Architectures
Wang et al.	[5]	IJCV (2023)	Partial	No	No	No	No	No
Wang et al.	[6]	IEEE TIV (2023)	Yes	No	No	No	No	No
Fan et al.	[8]	IEEE TIV (2024)	Partial	Yes	No	Partial	No	No
Wang et al.	[9]	IEEE TITS (2024)	Yes	No	No	No	No	No
Valverde et al.	[23]	Sensors (2024)	Partial	Partial	No	No	No	No
Shi et al.	[24]	CMC (2025)	Yes	Partial	No	No	No	No
Li et al.	[15]	IEEE TITS (2026)	Yes	Partial	Partial	No	Partial	No
Umar et al.	[28]	Adv. Eng. Inf. (2026)	No	Partial	No	No	No	No
This SRL	–	2026	Yes	Yes	Yes	Yes	Yes	Yes

TABLE II. RESEARCH QUESTIONS BASED ON IDENTIFIED GAPS

QR	Question statement	Addressed gap	Section of response
QR1	What are the main fusion strategies for 3D object detection?	Lack of 4D radar coverage in fusion pipelines and heterogeneous Vehicle-to-Everything strategies.	V-A
QR2	How do these techniques behave under various conditions (night/rain/fog, etc.)?	Absence of a review covering adverse conditions and adversarial security.	V-B
QR3	What datasets and metrics are used for evaluation?	Absence of V2X datasets and occupancy metrics.	V-C
QR4	Which emerging architectures are fundamentally transforming fusion?	Absence of reviews covering Mamba/SSM and limited coverage of LLMs.	V-D
QR5	What are the current limitations and the most promising future research trends?	Lack of certain important information.	V-E

computational constraints for embedded systems. Zhang et al. [22] point out that learning algorithms applied to urban driving must specifically contend with this diversity of conditions and sensors to remain reliable in practice. Fan et al. [8] focus on 4D mmWave radar, providing a comprehensive architecture and analyzing the physical properties of the sensors. However, the integration of 4D radar into tri-modal fusion architectures combining LiDAR, cameras, and radar remains largely unexplored. Its application in V2X scenarios and standardized comparisons with LiDAR under adverse weather conditions also require further investigation. Wang et al. [9] propose a clear study introducing an approach based on the level of fusion integration and analyzing the environment applications in smart cities. Li et al. [15] have recently presented a complete BEV survey that introduces the new architectures and topics. However, adverse weather conditions and adverse security have developed insufficiently. Valverde et al. [23] demonstrated a transverse view of modalities based on clear taxonomy and the historic analysis of the evolution of different methods, but the width of the perimeter limits the depth of the analysis and decreases the precision of quantitative comparisons between single-modal and fused approaches. Shi et al. [24] address these dependencies and analyze architectural trends. However, their selection methodology lacks strength due to the absence of a PRISMA protocol and Bibliometrix [29] analysis, while multi-object tracking and adversarial robustness are still unaddressed. Adnan et al. [25] specified their survey on multi-object tracking using clustering techniques such as DBSCAN and HDBSCAN, and tracking approaches like Kalman, IMM et DNN. Dal'Col et al. [26] extend the perspective to joint perception and trajectory prediction for autonomous driving, while Fawole and Rawat [27] survey recent advances in 3D detection for autonomous vehicles, confirming both the richness and the thematic dispersion of review efforts in this field.

Nevertheless, the study is limited to LiDAR only and does not cover the multi-sensor concepts and the transformers on tracking. The last survey was conducted by Umar and al [28]. It is a systematic survey using the Kitchenham methodology that includes 74 articles. A quality analysis and comprehensive coverage of multi-sensor depth estimation, including camera, LiDAR, radar, and fusion, were conducted. Nevertheless, this study remains focused only on depth estimation and does not address 3D detection, tracking, V2X cooperative perception, Mamba/LLM models, or adverse weather conditions.

In general, these surveys show significant progress in multimodal autonomous perception, but still reveal important gaps in the integration of new AI architectures, robustness in real-world environments, the security of autonomous systems, and distributed cooperative fusion. Table I recaps the comparative study.

This article presents a systematic review covering studies between 2022 and 2026. The main subject of this study is the advancement in 3D object detection in autonomous driving using multi-data sensor fusion architecture.

B. Purpose

This systematic review introduces five main objectives:

- identification and categorization of multi-sensor fusion approaches for 3D object detection in autonomous

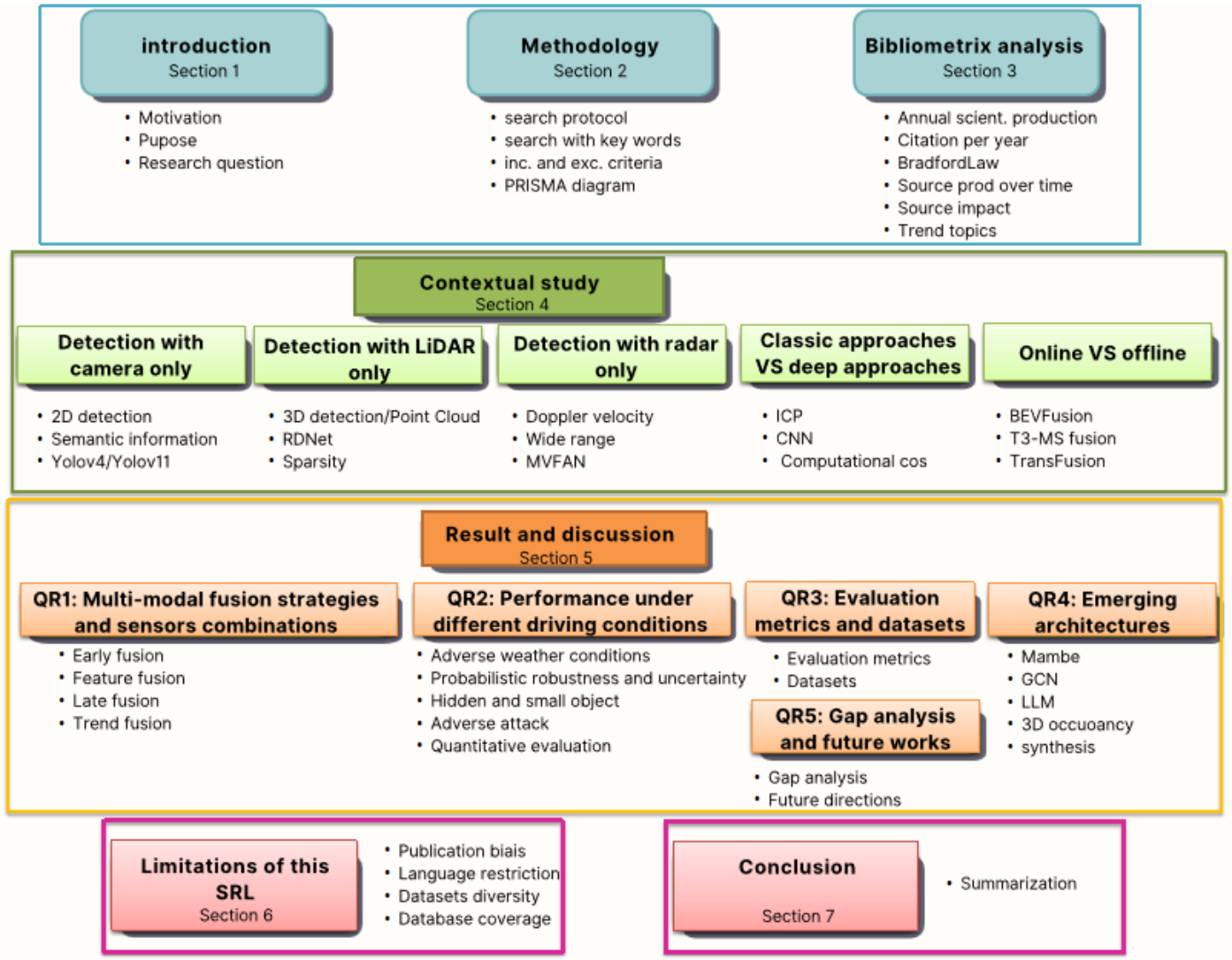


Fig. 1. The structure of review.

driving(2022-2026).

- Comparative analysis of the fusion strategy based on the fusion level and the algorithmic architecture.
- Dataset and metrics analysis based on previous studies.
- Identification research gaps and concrete directions proposition for future work.
- New architecture emergence synthesizes.

C. Research Questions

Based on a critical analysis of existing surveys and the identified gaps, we define five research questions that structure this entire review. These questions specifically target areas that are not covered or are insufficiently addressed in the existing literature. Table II represents the research questions.

In the rest of this review, articles search and selection methodology is presented in Section II. The Bibliometrix analysis is described in Section III. Contextual study is provided in Section IV. Discussion based on research questions is described in Section V. The limitations of this review is provided in Section VI. The conclusion is described in Section VII. The structure of this review is designed in Fig. 1.

II. Methodology

A. Search Protocol

This review is guided by the PRISMA protocol (Preferred Reporting Items for Systematic Reviews and Meta-Analyses). The research strategy is based on two complementary databases from Scopus and web of sciences(wos). In fact, PRISMA protocol provides a transparent, rigorous, and reproducible search strategy. It allows an evident description of all steps of the research process, from the identification of studies to their final selection specifying the databases consulted, the keywords used and the inclusion and exclusion criteria as demonstrating in the following sections.

B. Search Key Words and Filter on Databases

Detailed research on the field needs to carefully choose the key words in the search step. The key words used in the article search are described below:

- For Scopus: “multi-fusion” &“ multi sensor” & “3D object detection” & “autonomous vehicles”
- For WoS: “multi sensor” & “3D object detection” & “autonomous vehicles”

This research finalized by finding 318 articles from Scopus and 155 articles from Web of Sciences(WoS). Then, in the Scopus website, We firstly filtered with “ year of publication (2022-2026)”, and the number of articles is decreased to 430.

C. Inclusion and Exclusion Criteria

Identifying inclusion and exclusion criteria is a crucial step in a systematic review. It defines which studies should be retained or excluded according to the research objectives. These criteria ensure a consistent and objective selection process, reduce potential bias, and improve the reliability of the results. Inclusion criteria are used to retain studies that are relevant to the research questions, whereas exclusion criteria eliminate studies that fall outside the scope of the review, such as incomplete or duplicate studies.

The inclusion and exclusion criteria used in this study are presented in Table III. A total of 26 duplicate articles were removed using RStudio, and 187 papers were then excluded that are not open access leaving 154 articles. Biblioshiny was then used to filter the articles by language, and 15 articles published in languages other than English were excluded. Finally, a comprehensive review of the remaining articles was conducted, resulting in 110 articles begin retained for the final analysis. The selected studies focus on 3D object detection or tracking in autonomous driving and provide quantitative results for multi-sensor detection or tracking.

D. PRISMA Diagram

The PRISMA diagram is very important. In fact, It outlines the various steps followed by the researcher, such as the identification of references, the removal of duplicates, the screening of titles and abstracts, the assessment of full-text article eligibility, and the final inclusion of the selected papers. Fig. 2 presents the PRISMA graph.

III. Bibliometrix Analysis

Bibliometrix analysis is done thanks to the R-studio and biblioshiny command. The following subsections describe the analysis of the collected articles.

A. Annual Scientific Production

From 2023, the number of articles increase rapidly thanks to the BEVFusion paper, which introduces the fusion in BEV space as a dominant paradigm. The year 2025 presents the pic articles with more than 45 articles. Fig. 3 demonstrates the evolution of the number of articles per year.

B. Citation Per Year

The average number of citations decreases over time, which is expected given the different publication dates of the studies. Highly cited papers are mainly review articles and influential approaches, such as BEVFusion and TransFusion. In contrast, articles published in 2025 and 2026 have had less time to accumulate citations, which does not diminish their scientific contributions. Fig. 4 presents the average number of citations per year, with a notable increase in 2022 that coincides with the emergence of influential BEV-based fusion architectures such as BEVFusion.

C. Bradford Law

Based on Bradford law, the core of zone 1 presents mainly the dominant sources : Sensors (MDPI), IEEE Transactions on Intelligent transportation systems, IEEE Access, IEEE Robotics and Automation Letters, drones, world electric vehicle journal, and IEEE Transactions on Intelligent Vehicles. Fig. 5 demonstrates the classification zones based on BradfordLaw and the most relevant sources based on document number.

D. Source Production Over Time

Fig. 6 shows that Sensors is the dominant source producer from 2022 to 2026. On the other hand, the second dominant source is variable each year. Between 2022 and 2024, IEEE Robotics and Automation Letters wins to retain the second producer place, but in 2025 and 2026 it is back to the fourth producer. The attractive evolution is from IEEE Transactions on Intelligent transportation systems. In fact, it is translated from the fourth source with less than five articles in 2024 to the second one in 2026 with ten articles. In the same context, the world electric vehicle journal begins its way to produce articles on this field after 2023.

E. Source Impact

To improve the importance of the source in our search and based on Bibliometrix tools, we calculate the Local H-index of each source. The local impact H-index means that “x” articles from this source in the database(Scopus+WoS database) are cited at least “x” times on the studied articles(in the concatenate database).

Fig. 7 shows that the source “Sensors” is the source with the highest H-index impact that is equal to 6. As local impact H-index , 6 is an important value that improves the importance of these sources.

F. Trend Topics

Fig. 8 shows that the concepts 3D object detection and autonomous driving coincide in 2025 with its higher frequencies. In fact, it illustrates the novelty of the theme.

G. Summary

The previous subsections allow us to conclude the importance and tendance of our review. In fact, statics from the Bibliometrix analysis improve that the 3D object detection is a trendy subject and define the most important source in this field. As a conclusion, this Bibliometrix analysis helps us to

TABLE III. INCLUSION AND EXCLUSION CRITERIA

Criteria	Inclusion	Exclusion
Theme	3D object detection or tracking in autonomous driving	Map construction or trajectory planning
Language	Only English	Any other language
Date range	Published between 2022 and 2026	Before 2022
Source type	Peer-reviewed conference or journal articles	Non-peer-reviewed sources
Relevance	Quantitative results about multi-sensor detection or tracking	Articles using mono-modality (not critical)

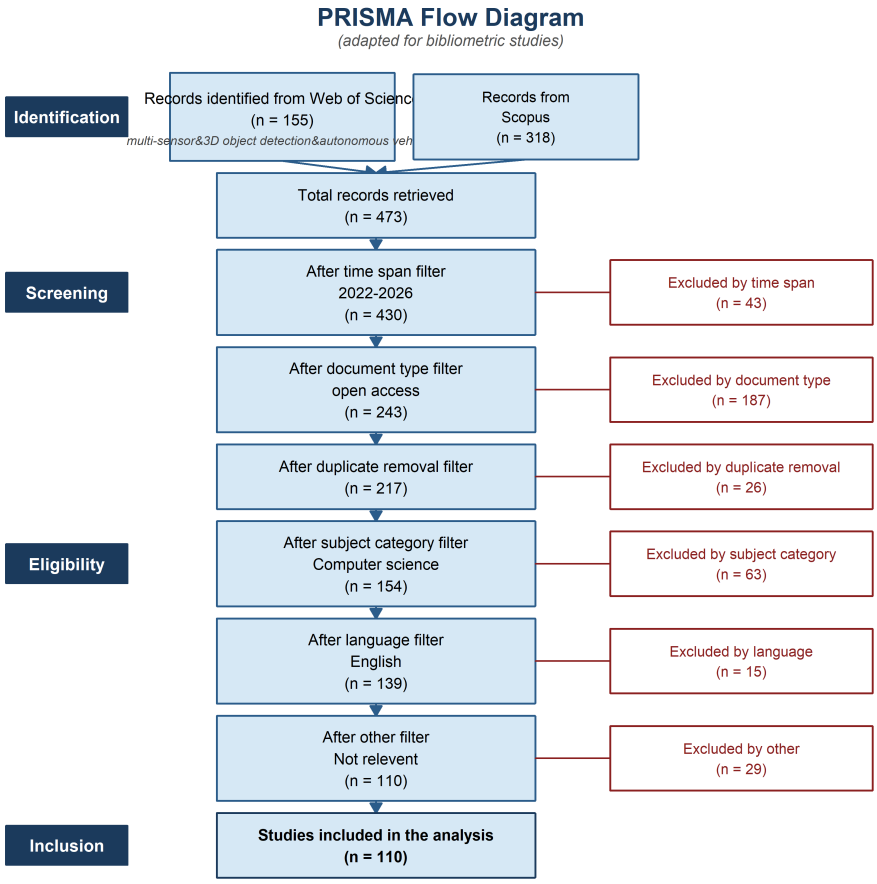


Fig. 2. PRISMA chart.

reflect on some notions related to 3D object detection theme which are defined in the research questions. In addition, it provides a look to the most important source of articles to enhance our study.

IV. CONTEXTUAL STUDY

The studies of the literature are interested in multi-sensor data fusion and depth estimation in autonomous driving. The comprehensive lecture of articles in the systematic review research allows us to conclude the the brilliance of the architectures such as transformer, Mamba, V2X and others. In addition, BEV is found in the literature as the basic of perception system and 3D detection. As a consequence, our research is dedicated to improve how these emergent architectures solve the BEV shortcomings. To satisfy this search, we need to start from the basic of multi-sensor fusion. The new approaches and architectures start from the mono-modality with classic architectures. The following sections will present the evolution of perception tools and approaches.

A. Detection Using Camera Only

In production, monocular methods are the most useful because they are the cheap ones. A 2D detection with high precision in real time is satisfied thanks to YOLOv4 [30] and YOLOv11 [31] architectures. However, 3D detection from the camera presents only a fundamental challenge. In fact,

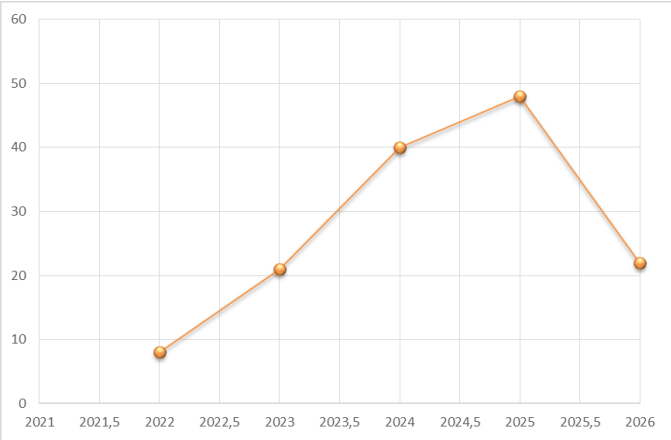


Fig. 3. Annual scientific production.

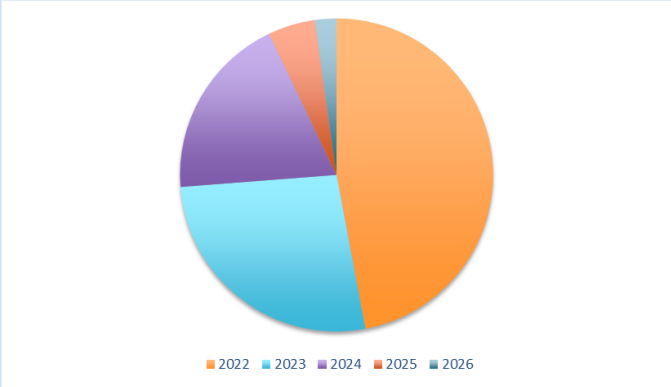


Fig. 4. Average citations per year.

numerous 3D scenes can coincide with the same 2D projection. To enhance the performance of detection with the camera only, studies propose exploring polar representation [33] and depth guidance [32]. Stable diffusion is used to outperform nocturnal images [34], the 3D object annotation via pseudo-LiDAR and the VLM validation [35] guide to an efficient annotation without LiDAR.

Even with all improvements, the monocular camera introduces several problems: depth, light and weather conditions, and insufficient cross-domain generalization for advanced levels of autonomous driving [10], [23]. The presented limits make the fusion with other modals crucial.

B. Detection Using LiDAR

Due to its capacity to directly measure the time-of-flight laser generating point clouds with high resolution and unchangeable under light conditions, LiDAR is presented as a reference to 3D detection [5], [6]. The approach RDNet [36] introduces a Rotate-ground-truth augmentation that allows to correct the unbalance of classes annotation (pedestrians are underrepresented) combined with a decoupled attention head to enhance the performance on KITTI and nuScenes datasets. Transformers in Multi-Feature Fusion Hierarchical-GCN(MUFFIN-HGCN) [37] allow pixel-point mapping without extrinsic parameters, combined with GCN hierarchic to

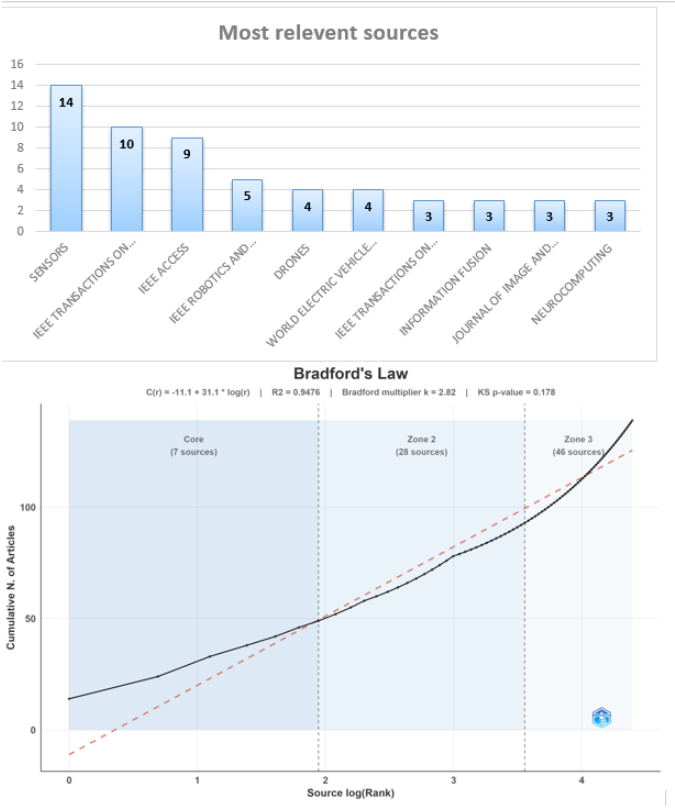


Fig. 5. BradfordLaw and most relevant sources.

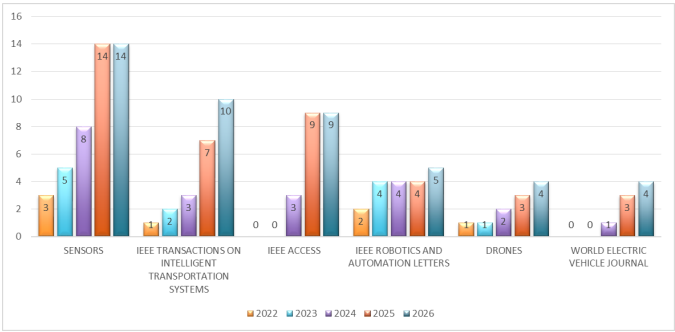


Fig. 6. Source dynamics.

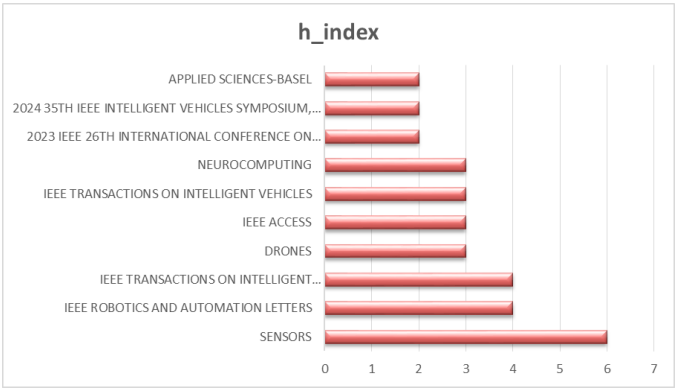


Fig. 7. Source impact.

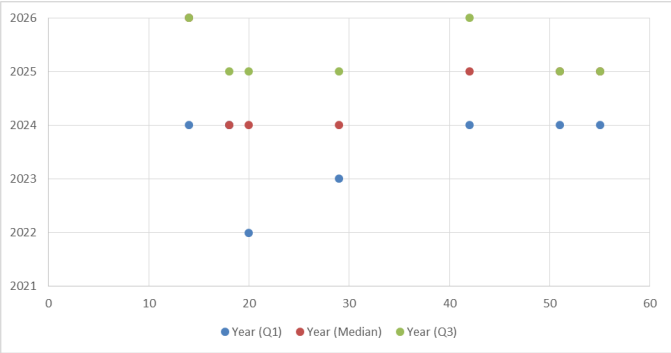


Fig. 8. Trend topics.

capture the non-Euclidean structure of point clouds, surpassing standard voxelizations. In real world deployment, LiDAR presents several deficiencies, including degradation under heavy precipitation (about 80% returns lost in dense fog) [42], sparsity and lack of semantic information for fine classification [5], [6].

C. Detection Using Radar Only

The unique physic complementarity is represented by the Millimeter radar. In fact, it does not degrade under intense precipitation, Doppler velocity measurement and provides a wide range of about 200-250m [8]. The 4D mmWave radar introduces the vertical dimension to provide a real spatial dimension. The SMIFormer [38] approach uses multi-view transformers for 3D detection with 4D radar. The Range-Angle-Doppler Net(RADNet) [39] method treats the 3D Range-Angle-Doppler tensor, avoiding information loss due to conversion to point clouds. Multi-View Feature Assisted Network for 4D Radar Object Detection (MVFAN) [13] introduces a multi-view architecture to enhance 3D detection using only 4D radar. Radar alone is still significantly inferior to LiDAR in spatial resolution, suffers from important speckle noise, and to false positives due to multi-path detection in dense urban environments. Precise object classification remains difficult without additional visual information [8], [12].

D. Classic Approaches VS Deep Learning Approaches

Classic approaches such as stereo triangulation, ICP for LiDAR,etc.. are interpretable and do not need labeled data, but they are noise-sensing, not very generalizable, and assume quasi-static conditions [8], [9]. Exploring large annotated corpora allows deep learning to change the 3D detection. CNNs, Transformers, and State-Space Models (Mamba) are the best for autonomous driving under various conditions, but a higher computational cost is the price of this enhancement [14], [15], [37]. Table IV represents the comparative table of sensor performance for 3D object detection tested on KITTI and evaluated with the metric Average precision, except the combination radar+camera due to the absence of radar in KITTI dataset.

E. Real-Time and Offline

The real-time and offline distinctions are crucial for deployment. Real-time models should treat each frame in less than 50 ms in em embedded calculators with energy vulnerabilities. They use light backbones such as MobileNet, PointPillars, and light ResNet-18, with quantification and pruning. T-3MS fusion [40] improves that an economical VLP-32C LiDAR and a 360° camera achieves state of the art without increased test-time. 360° DAMOT [41] marks energy efficiency with fast encoders and a portable Docker/ROS architecture. Offline models such as BEVFusion, TransFusion, and CoMCM [42] prioritize maximum accuracy for algorithmic validation but are computationally prohibitive in production. Tran et al.[43] propose T360Fusion, a multimodal 360° temporal fusion method based on Transformers that aims for a balance between accuracy and latency, while Hossain [44] applies lightweight multi-sensor fusion to short- and long-range obstacle detection for a delivery robot, illustrating the transferability of these architectures beyond the realm of autonomous road vehicles.

V. Results and Discussion Based on Reseach Questions

This section answers the five research questions defined from the gaps identified in Section I-C. Each subsection provides a structured, quantified, and critical analysis of the 110 articles.

A. QR1: Multi-Modal Fusion Strategies and Sensors Combinations

Fusion approaches categorize in four classes based on the integration level. Table V synthesizes the characteristics, advantages, and limitations of each method.

1) Early fusion: Early fusion combines raw data directly. PixelFusion [45] introduces a pixel fusion by projecting the point clouds of LiDAR on the image plan, allowing integration of fine information(texture/color) before the voxelization step. ExistenceMap-PointPillars [46] integrates pseudo-cards to PointPillars, improving robustness when data degradation. The Segmentation Guided Fusion Network(SGFNet) [65] uses a foreground segmentation head to guide the projection of the pixel-point, avoiding LiDAR sparsity artifacts. Zhao et al. [47] introduce a multi-scale voxel fusion module specifically optimized for small objects, achieving 73.81% AP_3D hard cars and 48.03% AP_3D hard pedestrians on KITTI. Other works refine the pixel-point fusion process itself: Tian et al. [48] employ singular value decomposition (SVD) to geometrically align LiDAR point clouds and camera images; Zuo et al. [49] propose a supervised cross-fusion method that reduces the need for dense annotations; and Yildiz et al. [50] combine real-time detection with semantic distance measurement using camera-LiDAR fusion. Chen et al. [51] propose CFFCNet, an object-center-guided feature completion network designed for more accurate localization and dimension estimation from LiDAR point clouds. Fig. 9 represents the process of early fusion methods.

2) BEV feature fusion: since 2022, the fusion of BEV features has been the dominant paradigm [14], [15]. In fact, BEVFusion combines camera and LiDAR characteristics in the same BEV space without depth problems. AttBEV [18] outperforms BEVFusion with the channel-spatial attention module CBAM, obtaining NDS = 67.95% (+2.63%) on nuScenes. MV2DFusion [16] generates independent semantic queries

TABLE IV. COMPARATIVE TABLE OF SENSOR PERFORMANCE FOR 3D OBJECT DETECTION

Modal	Advantages	Limits	Performance (tested on KITTI)
Camera	High resolution, rich semantic information and texture, and low cost	Depth errors, absence of direct 3D detection, and sensitivity to lighting and weather conditions	∼70–80% AP
LiDAR	3D precision and robust under lighting conditions	Degradation under precipitation, sparsity, and high cost	∼85–92% AP
4D Radar	Wide range, direct measurement of Doppler velocity, and robustness under adverse weather conditions	Low angular resolution and multipath noise	∼40–60% AP
LiDAR + Camera (BEV fusion)	Geometric and semantic complementarity	Critical calibration, high computational cost, and need for temporal synchronization	∼88–95% AP
Camera + Radar	Robustness under adverse conditions, semantic information, and Doppler velocity measurement	Modest 3D precision	∼55–70% NDS
LiDAR + Radar	Geometric complementarity and robustness under lighting and weather conditions	Pipeline is not yet standardized and datasets are scarce	∼87–92% AP

TABLE V. COMPARISON OF MULTIMODAL FUSION APPROACHES FOR 3D OBJECT DETECTION

Approach	Description	Advantages	Limitations	Examples (Refs.)
Early fusion	Fusion of raw sensor data before feature extraction	Rich multimodal information with low-level details preserved	Requires precise calibration and involves high computational cost	[45], [46], [47], [83], [82]
Feature fusion	Fusion of extracted features in a shared BEV space	Good performance–efficiency trade-off and improved robustness to miscalibration	Heterogeneous feature alignment and increased training complexity	[16], [17], [18], [59], [60], [66], [53], [67], [84], [85], [86], [87]
Late fusion	Combination of the outputs of independent models	Modularity, fault tolerance, and flexibility across different domains	Limited modeling of fine-grained cross-modal correlations	[61], [62], [63], [64], [88], [89], [90], [91]
Deep hybrid	Multi-level feature integration with adaptive fusion	High detection accuracy, context awareness, and adaptive multimodal integration	Requires substantial annotated training data and high computational resources	[10], [42], [12], [16], [20], [92], [93], [103]

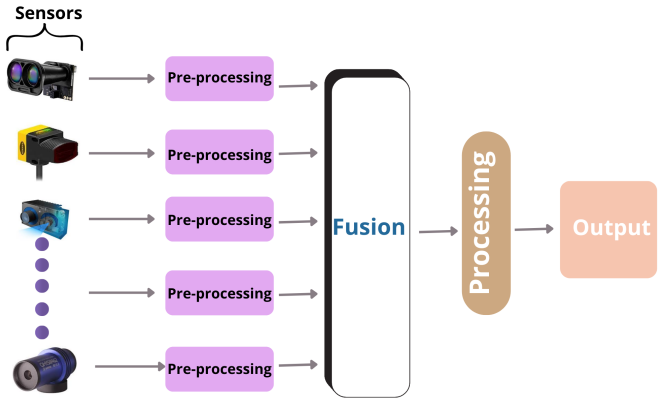


Fig. 9. Early fusion approach.

from images and point clouds, then combines them sparingly via cross-attention, achieving state-of-the-art performance on nuScenes and Argoverse2 [52]. Xu et al. [17] propose a Frustum-Aware Fusion Network with cross-attention, and a multi-modal alignment module providing 90,88% /84,47% /80,99% mAP_3D respectively for cars/pedestrians/cyclists on KITTI. Li et al. [66] provide a Deformable Cross-Modal Spatial Attention with learnable offsets, robust for calibration errors via the Height-Adaptive Aggregation strategy. BEV is enhanced by ECL3D [53] through the combination between Depth-Semantic Feature Enhancement (DSE) and Edge-guided Channel-Spatial Enhancement (ECSE) as part of Lift-Splat-Shoot without additional inference cost. RM²Occ[54] extends the BEV paradigm towards semantic 3D voxel occupancy prediction with bidirectional (from 2D to 3D and vice versa) reprojection and integration of the Inertial Measurement Unit(IMU), more suitable for trajectory planning. In the same context, Li et al.[55] propose Panoptic-VSNet, which leverages prior semantic knowledge to guide multimodal 3D panoptic segmentation, combining instance detection and dense scene understanding into a single BEV pipeline. Based on a similar principle of selectivity, Jacobson et al. [56] propose Center Feature Fusion, which first identifies relevant object locations in camera and LiDAR streams before fusing only a limited subset of associated features in BEV, thereby reducing computational cost compared to approaches that project the entire image. Liu et al.[57] enhance detection robustness in complex traffic through unified BEV feature alignment, while Wang et al. [58] propose TF-CMFA, which combines temporal fusion and cross-modal alignment for robust 3D detection in dynamic environments. The feature level fusion is described in Fig. 10.

3) Late and cooperative fusion: LCF3D [61] introduces a late cascade fusion (late + cascade) that provides two main functionalities: *Filtering the false positives of LiDAR*

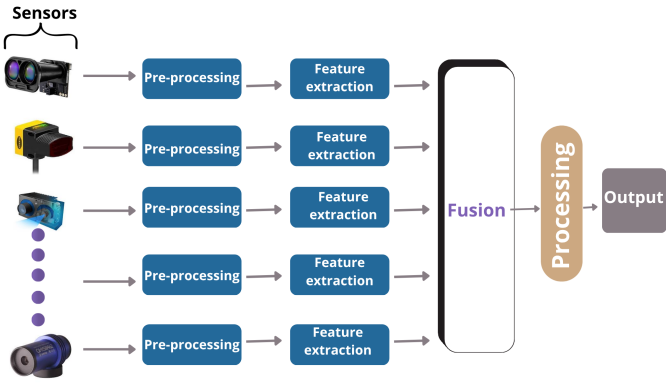


Fig. 10. Feature level fusion approach.

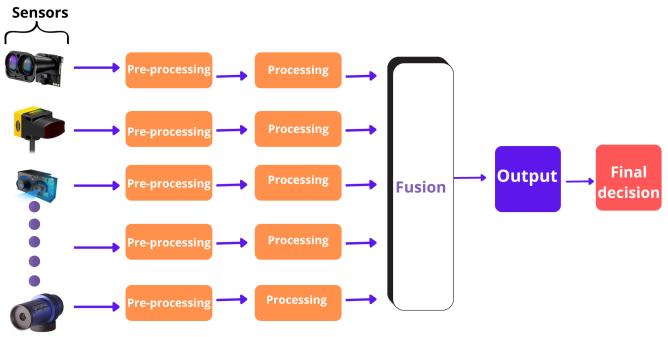


Fig. 11. Late fusion approach.

by the association of the 2D image camera detection and *from unmatched 2D detections, the 3D frustums retrieve of missed objects*. Its main objective is cross-domain. In fact, it allows for different configurations for sensors in training and testing. Without access to internal architectures, Fadili et al. [66] provide multi-source black-box cooperation that enables collaboration between agents from different manufacturers(crucial conditions for real deployment). CLF-BEVSORT [62] uses a Kalman filter for 3D multi-object tracking by combining camera and LiDAR in BEV. It offers a lightweight alternative to intensive deep learning methods. Naveen et al. [97] combine multiple sensors using deep learning for the onboard perception of autonomous electric vehicles. Dheepika and Umadevi [98] propose a hybrid approach combining 3D detection and tracking within a single deep learning pipeline. Regarding multi-object tracking, Cho and Kim [107] combine multi-positive contrastive learning with deep reinforcement learning for 3D LiDAR tracking, while Sochaniwsky et al. [108] employ a camera-LiDAR fusion framework designed for road traffic monitoring. These architectures also find applications in related fields: Liu et al. [109] and Bultmann et al. [110] apply similar multi-sensor fusion principles to drone landing and semantic segmentation; Rho et al .[105] to underwater 3D reconstruction; and Shang et al. [111] to obstacle avoidance for autonomous vehicles, demonstrating the broader transferability of these fusion techniques beyond the road traffic context alone. Yan et al. [100] enhance cooperative vehicle-infrastructure perception using enriched point cloud features and dedicated spatial attention, while Zimmer et al. [101] propose InfraDet3D, a multimodal detection architecture leveraging cameras and LiDAR mounted on road infrastructure. Yuan et al. [104]propose a keypoint-based deep feature fusion method for cooperative vehicle detection, serving as a foundational reference for this type of approach. The late fusion process is demonstrated in Fig. 11.

4) Trend fusion: Camera-radar: Since 2023, the combination of camera and radar has experienced significant growth. In fact, it is an economical and robust alternative [8], [64], [68], [69], [70], [71]. With spatial information and radar speed, CR3DT [64] extends BEV detection and camera only methods. It provides +5,3% mAP and +14,9% average multi object tracking accuracy (AMOTA) in nuScenes. To improve detection and tracking in bad weather, NeXtFusion [68] uses attention mechanisms. The Dual-branch network allows to design interactions between camera and radar with point cloud offset compensation. Xiao et al. [70] combine perspective view fusion for depth estimation and BEV fusion for final detection. MLF-4DRCNet [71] introduces a 4D radar + camera multi-level fusion with cross-attention on nuScenes. The LiDAR and radar fusion is possible and offers a robust system under adverse conditions and a better range as demonstrated in [72] but it lacks semantic information. Qi et al. [112] proposes FusionBev, an architecture that also combines LiDAR and 4D radar in BEV space, confirming the growing relevance of this sensor combination for robust 3D detection. Li et al. [99], for their part, guide radar vision fusion using multi-scale region proposals, whereas Ruddat et al. [102] specifically address the detection of vulnerable road users through fusion that is robust to 4D radar sparsity. In the same context, Khatab et al. [106] specifically evaluate the detection of vulnerable objects using a dedicated multi-sensor system.

B. QR2: Performance Under Different Driving Conditions

1) Adverse weather conditions: To face adverse weather conditions, ContextualFusion [73] proposes a fusion with convolutional gate. As a result, ContextualFusion performs better than the standard BEVFusion providing +11,7% mAP at night and +8,3% in rain in nuScenes. Depending on detected weather conditions, SAMFusion [74] proposes to dynamically weight the modalities. The adaptive sensor fusion demonstrates significant gains in rain, fog, and night. Tian et al. [75] demonstrate that the LiDAR-radar combination with voxel augmentation and BEV synchronization significantly improves 3D detection in dense fog in Oxford Radar RobotCar. HRFuser [76] is evaluated on nuScenes DENSE (snow, fog, rain), demonstrating the robustness of multi-resolution radar fusion in nighttime conditions. Fast All-day [77] fuses LiDAR and a visible camera for detection in all weather conditions. More recently, Wang et al. [113] proposed SWG-Fusion, which leverages a vision-language model (VLM) to dynamically guide multimodal fusion in BEV based on the severity of detected weather conditions, illustrating the growing integration of language models into mechanisms for robustness against adverse conditions.

2) Probabilistic robustness and uncertainty: An adaptive uncertainty quantification for the Kalman filter is proposed by Ulreich et al. [78]. In fact, the proposed method allows to dynamically estimating the noise measure based on the condition, making the tracking more robust face to sensor degradation. Yang et al. [79] introduce Evidential Fusion based on Dempster-Shafer theory, separately treating epistemic and random uncertainty for a dynamic weighting of modalities according to their instantaneous reliability.

3) Hidden and small objects: The detection of hidden and small objects is a major and crucial challenge. In fact, the extraction of LiDAR features in probable occlusion regions from 3D frustums defined by 2D detections is guided by a Spatial Attention Frustum [80]. Adaptive Multiscale Attention Feature Aggregation is developed with contextual priors in occlusion regions [81]. Zhao et al. [47] target small objects thanks to a multi-scale voxel fusion module with three granularities.

4) Adverse attack: El-Fatyany [94] identifies a fundamental structural vulnerability: multi-sensor fusion pipelines can be manipulated by physical patch attacks inserted into the camera modality, deceiving the 3D detector despite an intact LiDAR. The proposed solution based on Unet segmentation is a first step to face this type of attack.

This gap constitutes an unquantified operational risk for the public deployment of autonomous vehicles.

5) Quantitative evaluation: TableVI demonstrates the quantitative evaluation of the methods of the articles in this state-of-the art.

Entries are grouped by dataset and that cross-row comparisons across different metrics/datasets are illustrative only, not statistically equivalent.

C. QR3: Evaluation Metrics and Datasets

Evaluation metrics and dataset selection are a crucial step in algorithm evaluation and comparison study between different methods

1) Evaluation metrics: Metrics allows a comparative study between different systems. The most useful metrics are:

- mean Average Precision (mAP): the main metric for 3D detection, it represents the average precision for all classes and thresholds IoU (0.5 for pedestrians and 0.7 for cars in KITTI dataset)
- (nuScenes Detection Score (NDS): the standard of leaderboard nuScenes, it is a composite score combining the mAP with five kinematic attributes.
- Higher Order Tracking Accuracy (HOTA): detection, association and localization are the three fundamentals components of object tracking, the HOTA score is designed to balance them.
- Average Multiple Object Tracking Accuracy (AMOTA): the standard metric of nuScenes for the multiple object tracking dataset.
- AP_BEV/AP_3D: precision in Bird's Eye View and in 3D on KITTI with IoU threshold 0.5/0.7.
- Intersection over Union (IoU): the main metric used in computer vision to measure the overlap between two geometric regions. The IoU is used to evaluate the accuracy of object detection, image segmentation, and object tracking models.
- Identity Switches (IDS): represents the number of identity switches indicating the quality of temporal association in tracking.

2) Datasets: Datasets are crucial for autonomous driving systems. Its variety allows to test the systems in different conditions, to evaluate its performance and to compare between different modals. Each dataset has its own characteristics such as the equipments of the vehicle (sensors), scenes, classes (pedestrians, cars, roads,..). TableVII represents an overview of the datasets used in multi-sensor fusion.

D. QR4: Emerging Architectures

Each modality is encoded separately using a CNN or Transformer, and the resulting features are projected into a shared BEV space and fused through concatenation or cross-attention. This represents the dominant paradigm for multi-sensor fusion in 3D detection. Nevertheless, this paradigm has several limitations when moving beyond standard conditions, particularly in non-Euclidean point clouds, multi-vehicle cooperation, complex semantic contexts, and direct trajectory planning.

To address these limitations, emerging architectures, including Mamba, GCNs, LLMs, and 3D occupancy-based models, have been developed. This section identifies the limitations of conventional architectures and discusses how emerging architectures address these challenges. Table VIII summarizes the main limitations and the corresponding architectures proposed to address them.

1) From local fusion to sequential cooperative fusion: Mamba/SSM:

a) The addressed problems: Cooperative multi-vehicle perception requires combining BEV representations from multiple moving agents. The representations are captured at marginally different timestamps, with point cloud densities that vary based on each vehicle's relative position. Transformers address this fusion through cross-attention between the agents' BEV representations. However, two structural issues render this approach suboptimal in this precise context.

- first problem: The quadratic complexity $O(n^2)$ of self-attention: In multi-vehicle, the sequence of BEV tokens for N vehicles grows linearly with N, while memory and computation time grow quadratically.
- Second problem: The non-causal aspect of attention: Each vehicle transmits its BEV state at regular intervals. Standard attention treats all tokens as temporally equivalent, ignoring the causal structure of this successive communication. The context transmitted by the vehicle at time t–1 should influence the interpretation of the transmission at time t, yet attention does not model this type of sequential dependency.

To resume, the perception of V2V is based on temporal state of BEV of N agents. Transformers ignored sequential

TABLE VI. QUANTITATIVE EVALUATION OF MULTIMODAL 3D OBJECT DETECTION METHODS

Ref.	Modalities	Datasets	Results	Test conditions
[17]	LiDAR + Camera	KITTI	90.88%, 84.47%, and 80.99% AP_{3D} for the Car category	Easy, Moderate, and Hard settings
[17]	LiDAR + Camera	nuScenes	89.2% and 64.9% AP_{3D} for Pedestrian and Bicycle detection	Moderate setting
[18]	LiDAR + Camera	nuScenes	67.95% NDS	Standard evaluation
[16]	LiDAR + Camera	nuScenes	76.7% NDS and 74.5% mAP	Standard evaluation
[75]	LiDAR + Radar	Oxford RobotCar	Accuracy rates of 93.2%, 92.5%, and 88.7% at IoU levels of 0.5, 0.65, and 0.8, respectively	Foggy conditions
[74]	Camera + LiDAR + Radar	Seeing Through Fog	Improvement of 17.2 AP points for pedestrians in dense fog and 15.62 AP points in heavy snow at long range	Adverse weather conditions
[73]	LiDAR + Camera	AdverseOp3D	6.2% mAP improvement over state-of-the-art methods	Adverse weather conditions (rain and night)
[64]	Camera + Radar	nuScenes	5.3% mAP improvement and 14.9% AMOTA improvement	Standard evaluation
[61]	LiDAR + Camera	KITTI / nuScenes	Best or second-best performance among the evaluated methods	Different evaluation parameters
[40]	LiDAR + Camera	nuScenes	74.9% NDS and 72.8% mAP	Class imbalance
[47]	LiDAR + Camera	KITTI	73.81% 3D AP for cars and 48.03% 3D AP for persons	Hard level for cars and persons (small objects)
[42]	LiDAR V2V	OPV2V	0.704 AP@0.5 and 0.495 AP@0.7	Cooperative perception
[10]	LiDAR + Camera	OPV2V and DAIR-V2X	State-of-the-art performance	Heterogeneous cooperative perception tasks
[20]	LiDAR-based single vehicle + infrastructure camera	V2XSet-V2I	0.292 improvement in AP@0.5 and 0.289 improvement in AP@0.7	Infrastructure and collaborating vehicles

TABLE VII. COMPARISON OF DATASETS USED FOR MULTIMODAL 3D OBJECT DETECTION

Dataset	Year	Scenes	Annotated frames	Classes	Sensors
nuScenes	2019	1000	40K	23	Cameras + LiDAR + Radar
KITTI	2012	50	15K	8	Cameras + LiDAR
AdverseOp3D	2024	100	4K	23	Cameras + LiDAR + Radar
VoD	2022	∼8700	∼8700	13	Camera + Radar + LiDAR
K-Radar	2022	62	35K	5	Camera + LiDAR + Radar
Oxford Radar RobotCar	2019	32	240K	Focus on zone geometry	Camera + LiDAR + Radar + GPS/INS
Seeing Through Fog (nuScenes DENSE)	2020	>10K km recorded in real test scenarios	1600	4	Camera + LiDAR + Radar
Waymo Open Dataset	2019	1150	12.5M	4	Camera + LiDAR
OPV2V	2021	73	12K	Vehicles	Camera + LiDAR
DAIR-V2X	2022	Not reported	71,254	15	Camera + LiDAR

TABLE VIII. LIMITATIONS OF BEV TRANSFORMERS AND CORRESPONDING ARCHITECTURES FOR MULTI-SENSOR FUSION

BEV Transformer Limitations	Effects on Multi-Sensor Fusion	Corresponding Architecture
Quadratic complexity ($O(n^2)$)	High computational cost when processing long BEV sequences and multi-vehicle scenarios	Mamba [42]
Non-causal BEV representation	Agent cooperation and temporal dependencies are not explicitly captured by the attention mechanism	Mamba [42]
Processing on a Euclidean grid	Voxelization can lose fine-grained spatial relationships in non-Euclidean point clouds	GCN [37]
Limited semantic context fusion	Limited ability to reason about scene-level semantic context and assess interactions between heterogeneous agents	LLM [12]
Discrete representation	Limited representation of the continuous space between objects	3D Occupancy [54]

causal structure and suffer from quadratic complexity, making them very expensive for real-time multi-agent pipelines.

b) How Mamba face to these problems?: Mamba is a selective State-Space Model that models sequences through a segmented continuous-time state equation. Its operation in the context of cooperative fusion relies on two main aspects.

- First, linear complexity ($O(n)$): Due to a parallel scan algorithm, Mamba processes sequences in linear time. For a sequence of BEV tokens of length n, the memory cost is ($O(n)$), rather than ($O(n^2)$). This property makes it possible to perform fusion over long multi-agent BEV sequences for on-board deployment.
- Second, contextual selection: The parameters of the state equation are input-dependent and change according to the content of each BEV token. This property allows Mamba to dynamically select relevant information from previous agents to interpret the current state.

In conclusion, CoMCM [42] proposes two Mamba modules for cooperative fusion. The Contextual Clustering Mamba module scans each agent's BEV tokens in two details: global ,the agent's full sequence, and local ,semantic clusters within each BEV map. The Collaborative Mamba in the BEV module then merges the hidden states of N agents into a unified cooperative BEV representation via a multi-agent scan. The result is state-of-the-art performance on OPV2V, with inference latency suitable for real-time on-board deployment.

2) From voxelization to non-Euclidean structure: GCN:

a) The addressed problem: A Point cloud is a set of points irregularly distributed of 3D space without regular grid. Before treating it with CNN or Transformer, most of architectures such as PointPillars, VoxelNet, and BEVFusion convert the point clouds in a regular representation. This voxelization resolves the problem of irregularity, but loses fundamental structural information. Standard voxelization erases non-Euclidean spatial relationships between LiDAR points, such as connectivity, local surface properties, and curvature. CNNs and Transformers applied to voxel grids cannot rebuild these lost relationships. Consequently, detection accuracy for small objects and fine contours is critical.

b) How GCNs respond to this problem?: A hierarchical Graph Neural Network clearly builds a graph from the LiDAR point cloud, where each point is a node, and edges encode spatial relationships between neighboring points. Messages passing between nodes enable the aggregation of local information and the computation of feature representations that reflect the actual structure of points cloud. The key element is hierarchy: the hierarchical GCN creates multiple scales of graph representation, ranging from the finest (relationships between individual points) to the coarsest (relationships between object regions). At each level, a graph pooling operator reduces the number of nodes while preserving important structural relationships. This multi-level approach is conceptually similar to multi-scale CNNs, but operates on graph structures rather than grids.

In fact, It outperforms standard voxelization architectures on KITTI and nuScenes, especially for small objects.

3) From geometric fusion to contextual semantic fusion: LLM:

a) The addressed problems: All multi-sensor fusion architectures operate on the fundamental geometric and statistic level. They detect objects by associating visual and geometric features with predefined categories, using supervised learning on a closed dataset. The major limit of this approach is designed in three situations.

- First, contextual reasoning between agents is impossible. In a V2X system where a vehicle shares its detections with a pedestrian on the street, a geometric fusion system cannot reason about the meaning of this configuration: is it a pedestrian about to cross? A cyclist stopping? CNNs and geometric Transformers do not possess semantic understanding of human behavior that can reasoning these scenes.
- Second, open-vocabulary classification is beyond the reach of current systems. In fact, any object not present in the training categories is ignored or misclassified.
- Third, semantic evaluation of risk. It requires a reasoning beyond the simple detection. This needs a causal model of the world that geometric architectures cannot build.

b) How LLM deal with these problems?: The UAV-vehicle is a cooperative framework driven by LLM(GPT-4) proposed by de Curtò & de Zarzà [12]. In fact, the language model merges and interprets multi-modal data from heterogeneous points of view, providing a semantic risk assessment. Although based on simulated data, this study opens up a fundamentally new path for contextual semantic fusion that goes beyond purely geometric approaches.

4) From object detection to spatial understanding: 3D occupancy:

a) The addressed limitations: 3D bounding box detection has been the standard representation for multi-sensor fusion systems since PointPillars and VoxelNet. This representation is standardized, but it presents three major limitations for trajectory planning.

- First, Bounding boxes do not represent free space. For safe trajectory planning, the module needs to know not only the location of the object but also where the navigable space lies.
- Second, Bounding boxes are inappropriate for objects with irregular forms.
- Third, Bounding boxes need a precise temporal association for tracking. The data-to-track association problem must be solved at all times.

This association problem is several under the conditions of occlusion, object crossings, or missing detections.

b) How 3D occupancy face these limitations?: Reprojection Multi-Task Multi-Sensor [54] is the first 3D occupancy network to integrate LiDAR-camera fusion within a multi-task framework. Its architecture comprises three modules:

TABLE IX. SYNTHESIS OF THE IDENTIFIED RESEARCH GAPS

Gap	Description	Impact
Adverse-weather benchmarks	Lack of standardized real-world datasets covering adverse weather conditions (night, rain, fog, etc.)	Critical
Adversarial security of multi-sensor fusion	Only one article investigates adversarial attacks in multi-sensor fusion [25]	Critical
Computational efficiency	Most state-of-the-art methods are too computationally demanding for on-board systems	High
Heterogeneous cross-manufacturer V2X	Multi-agent systems equipped with heterogeneous sensor configurations remain insufficiently studied	High
Multi-sensor 3D occupancy	Few alternatives to bounding-box representations are available for downstream planning	Medium
Cross-domain generalization	Models often exhibit limited transferability to new environments and geographical regions [96]	High
Robust online calibration	Few approaches remain robust under dynamic sensor miscalibration [95]	High
Mamba/LLM	Promising approaches, but their performance has not yet been extensively validated on large-scale real-world datasets	Medium

- A dual-branch encoder is represented in the LiDAR branch via voxelization and in the camera branch via Lift-Splat-Shoot
- A BEV fusion module featuring bidirectional 2D/3D reprojection and IMU data integration for temporal consistency
- A multi-task head that predicts the semantic voxel occupancy grid and standard 3D bounding boxes at the same time

Compared to previous monocular camera occupancy methods, the results demonstrate an improvement in voxel IoU on nuScenes, while maintaining standard 3D detection performance.

5) Synthesis: Emerging architectures are fundamentally transforming multi-sensor fusion not through incremental improvements, but through a paradigm shift over four dimensions: the temporal scalability of multi-agent cooperation [42], the topological fidelity of LiDAR encoding [37], contextual semantic scene reasoning [12], and the very nature of the fusion output representation [54]. Each addresses a structural limitation that dominant BEV Transformers cannot resolve without fundamentally adjusting their paradigm.

E. QR5: Gap analysis and future works

1) Gap analysis: This study allows us to introduce eight gaps. These gaps define the most promising opportunities for future work and establish the field's priority areas for development. Table IX synthesis the identified gaps.

2) Future direction:

- Build self-calibrating models that adapt to sensor drift in real time.
- real-world datasets with standardized weather conditions subtile for comparison.
- Light onboard architectures.
- Developing robust systems against adversarial attacks.
- Lightweight 3D occupancy.
- Mamba and LLMs for contextual fusion.

VI. LIMITATIONS OF THIS STUDY

This systematic literature review was conducted according to rigorous academic protocols. However, several limitations must be acknowledged for the spirit of transparency and the credibility of the results.

A. Publication Bias

The articles studied are peer-reviewed publications. This excludes unpublished work, industrial technical reports, and preprints from major companies. The excludes works could offer valuable insights into fusion systems under real-world operating conditions.

B. Language Restriction

Only articles in English were considered. Relevant studies published in Chinese, Japanese, or Korean may have been excluded.

C. Datasets Diversity

The predominance of KITTI and nuScenes, which can bias evaluations toward specific geographic and climatic conditions. This can affect driving in rural environments and developing countries.

D. Database Coverage

Although Scopus and WoS are the most comprehensive reference databases, some relevant articles may have been missed due to indexing variations or alternative terminology.

VII. CONCLUSION

This systematic literature review, conducted in accordance with the PRISMA protocol, analyzed 110 articles published between 2022 and 2026, rigorously selected from Scopus and Web of Science.

The results confirm that multi-sensor fusion provides significantly superior accuracy, robustness, and reliability compared to single-sensor approaches. Among the various strategies, feature fusion in BEV space with cross-attention mechanisms or deformable transformers has become the dominant

paradigm. Late fusion introduces a cross-domain generalization advantage, and cooperative V2X hybrid deep fusion opens up significant prospects.

Architecture has been enhanced since 2024. The major architectural contributions are: Mamba state-space models for Bird's Eye View Vehicle to vehicle cooperation, fusion based on probabilistic uncertainty quantification, LLMs for multi-agent semantic fusion, multi-sensor, multi-task 3D occupancy, and heterogeneous cross-modality V2X cooperation. 4D mmWave radar has established itself as an essential third complementary modality.

The identified gaps define a clear research docket:

- A unified adverse weather benchmark based on real world data.
- Adversarial security for multi-sensor fusion.
- Lightweight architecture for embedded deployment.
- Cross-domain generalization.
- Robust online calibration.

These challenges conduct the transition from academic results to safe, certifiable, and large-scale deployed autonomous driving systems.

ACKNOWLEDGMENT

The authors acknowledge that a large language model was used only to correct grammatical and syntactic errors and to improve the fluency and readability of the manuscript. The scientific content and ideas remain those of the authors.

REFERENCES

[1] A. Sahbani, NARMA-L2 Neuro controller for speed regulation of an intelligent vehicle based on image processing techniques, 21st Saudi Computer Society National Computer Conference, NCC 2018, doi.org/10.1109/NCG.2018.8593173.

[2] J. Cao, Y. Han, D. Wang, W. Zhao, Y. Zhao, and Q. Hou, "3D object detection method based on BEV feature fusion", Qiche Gongcheng/Automotive Engineering, vol.48, no.1, pp.80–90, 2026.

[3] X. Li, "Transformer-based fusion of LiDAR and polarimetric imaging for autonomous driving", in Proc. SPIE, vol.14064, 2026.

[4] Road traffic injuries, Accessed: September 13, 2026, Available at: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[5] Y. Wang et al., "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey", in International Journal of Computer Vision 131, 2122–2152, 2023, doi: 10.1007/s11263-023-01784-z.

[6] L. Wang et al., "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy", in IEEE Transactions on Intelligent Vehicles, vol. 8, no. 7, pp. 3781-3798, July 2023, doi: 10.1109/TIV.2023.3264658.

[7] R. Sebai, A. Sahbani and T. Bejaoui, "Robust Multi-Sensor Fusion Architecture Tested on CARLA Sim", 2026 International Conference on Smart Applications, Communications and Networking (SmartNets), Rome, Italy, 2026, pp. 1-6, doi: 10.1109/SmartNets69662.2026.11604736.

[8] L. Fan et al., "4D mmWave Radar for Autonomous Driving Perception: A Comprehensive Survey", in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 4, pp. 4606-4620, April 2024, doi: 10.1109/TIV.2024.3380244.

[9] X. Wang, K. Li and A. Chehri, "Multi-Sensor Fusion Technology for 3D Object Detection in Autonomous Driving: A Review", in IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 2, pp. 1148-1165, Feb. 2024, doi: 10.1109/TITS.2023.3317372.

[10] Z. Li et al., "CD-HCP: A Heterogeneous Cooperative Perception Framework Utilizing Cross-Modality Dual-Attention", in IEEE Transactions on Intelligent Transportation Systems, vol. 27, no. 6, pp. 6787-6799, June 2026, doi: 10.1109/TITS.2026.3670252.

[11] Y. Zhang et al., "CooPercept: Cooperative Perception for 3D Object Detection of Autonomous Vehicles", in Drones Journal, 2024, Volume 8, Issue 6, doi: 10.3390/drones8060228.

[12] J. De Curtò and I. De Zarzà, "LLM-Powered Cooperative Perception Framework for Mixed UAV-Vehicle Platoons", 2025 10th International Conference on Fog and Mobile Edge Computing (FMEC), Tampa, FL, USA, 2025, pp. 282-289, doi: 10.1109/FMEC65595.2025.11119241.

[13] Q. Yan, & Y. Wang, "MVFAN: Multi-view Feature Assisted Network for 4D Radar Object Detection", In Neural Information Processing: 30th International Conference, ICONIP 2023, Changsha, China, November 20–23, 2023, Proceedings, Part IV, Pages 493 - 511, doi: /10.1007/978-981-99-8070-3_38.

[14] H. Li et al., "Delving into the Devils of Bird's-eye-view Perception: A Review, Evaluation and Recipe", in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 46, NO. 4, APRIL 2024.

[15] H. Li, Y. Zhao, J. Zhong, B. Wang, C. Sun and F. Sun, "Delving Into the Secrets of BEV 3D Object Detection in Autonomous Driving: A Comprehensive Survey", in IEEE Transactions on Intelligent Transportation Systems, vol. 27, no. 1, pp. 119-144, Jan. 2026, doi: 10.1109/TITS.2025.3624830.

[16] Z. Wang, Z. Huang, Y. Gao, N. Wang and S. Liu, "MV2DFusion: Leveraging Modality-Specific Object Semantics for Multi-Modal 3D Detection", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 48, no. 1, pp. 609-623, Jan. 2026, doi: 10.1109/TPAMI.2025.3609348.

[17] J. Xu, K. Xiao, W. Ji and C. Li, "A Frustum-Aware Fusion Network With Cross-Attention for Multi-Modal 3D Detection," in IEEE Access, vol. 13, pp. 200777-200790, 2025, doi: 10.1109/ACCESS.2025.3637156.

[18] N. Zhang et al., "AttBEV: enhancing multi-modal 3D object detection with CBAM attention in BEVFusion for autonomous driving", in IEEE robotics and automation letters Journal, Feb 2025, vol. 11, num. 2,DOI:10.1109/LRA.2025.3641130.

[19] J. Zhong et al., "Leveraging Temporal Contexts to Enhance Vehicle-Infrastructure Cooperative Perception", 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), Edmonton, AB, Canada, 2024, pp. 915-922, doi: 10.1109/ITSC58415.2024.10920140.

[20] X. Zhou, C. Wang, Q. Xie and T. Qiu, "V2I-Coop: Accurate Object Detection for Connected Automated Vehicles at Accident Black Spots With V2I Cross-Modality Cooperation", in IEEE Transactions on Mobile Computing, vol. 24, no. 3, pp. 2043-2055, March 2025, doi: 10.1109/TMC.2024.3486758.

[21] Y. Liao et al., "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3292-3310, 1 March 2023, doi: 10.1109/TPAMI.2022.3179507.

[22] Sh. Zhang et al., "Learning-based algorithms with application to urban scene autonomous driving", in Artificial Life and Robotics Journal, Volume 28, pages 244–252, 2022, doi: 10.1007/s10015-022-00813-3.

[23] M. Valverde et al., "A Survey of Deep Learning-Based 3D Object Detection Methods for Autonomous Driving Across Different Sensor Modalities". Sensors. 2025; 25(17):5264. https://doi.org/10.3390/s25175264.

[24] P. Shi et al., "Research Progress on Multi-Modal Fusion Object Detection Algorithms for Autonomous Driving: A Review", in Computers, Materials and Continua Journal, Volume 83, Issue 3, 19 May 2025, Pages 3877-3917, doi: 10.32604/cmc.2025.063205.

[25] M. Adnan et al., "Systematic and Comprehensive Review of Clustering and Multi-Target Tracking Techniques for LiDAR Point Clouds in Autonomous Driving Applications", in Sensors Journal 23, no. 13: 6119, 2023, doi: 10.3390/s23136119.

[26] L. Dal'Col et al., "Joint Perception and Prediction for Autonomous Driving: A Survey", in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 12, pp. 21427-21452, Dec. 2025, doi: 10.1109/TITS.2025.3607743.

[27] O.A. Fawole et al., "Recent Advances in 3D Object Detection for Self-

Driving Vehicles: A Survey", in AI Journal 5, no. 3: 1255-1285, 2024, doi: 10.3390/ai5030061.

[28] M. Umar et al., "Advances in multi-sensor fusion for depth estimation in autonomous vehicles: A comprehensive survey", in Advanced Engineering Informatics, September 2026, Article: 104657, Volume 74, Part A, doi: 10.1016/j.aei.2026.104657.

[29] Aria, M., & Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. Journal of Informetrics , 11(4), 959-975.

[30] R. Murendeni et al., "Using a YOLO Deep Learning Algorithm to Improve the Accuracy of 3D Object Detection by Autonomous Vehicles", in World Electric Vehicle Journal. 2025; 16(1):9,doi:/10.3390/wevj16010009.

[31] M. Simeonov, P. Kamencay and M. Dado, "Evaluating YOLOv11's Role in Robust Real-Time Object Detection for Autonomous Driving", 2025 35th International Conference Radioelektronika (RADIOELEKTRONIKA), Czech Republic, 2025, pp. 1-5, doi: 10.1109/RADIOELEKTRONIKA65656.2025.11008422.

[32] Y. He, "3D Object Detection via Multi-Sensor Fusion and Deep Guidance", in Applied and Computational Engineering Journal,DOI: 10.54254/2755-2721/2026.GU33954.

[33] J. Tang, Y. Hou, J. Liu and B. Yu, "PRTF: Polar Space Represented Multi-View 3D Object Detection With Temporal Fusion Enhancement", in IEEE Transactions on Intelligent Transportation Systems, vol. 27, no. 1, pp. 1538-1549, Jan. 2026, doi: 10.1109/TITS.2025.3633448.

[34] Y. zhang et al., "A stable diffusion-based method for low-light image enhancement in autonomous driving",in Intelligent Journal Machine Learning & Cyber. 17, 238 2026,doi: 10.1007/s13042-026-03053-0.

[35] Ma. Yalong, et al, "Efficient 3D object annotation via vision-derived pseudo-LiDAR and Vision Language Model (VLM) validation", in Transportation Research Part C: Emerging Technologies, Volume 182, 2026, 105429, ISSN 0968-090X, doi: 10.1016/j.trc.2025.105429.

[36] Z. Xia et al., "RDNet: Rotate-Groundtruth Augmentation and Decoupled Attention HEAD for 3D Object Detection", in IEEE Transactions on Big Data, vol. 12, no. 2, pp. 389-402, April 2026, doi: 10.1109/TBDATA.2025.3624947.

[37] C. Liu, et al, "MUFFIN-HGCN: Multi-Feature Fusion Hierarchical-GCN for 3D Object Detection", in IEEE Transactions on Vehicular Technology, vol. 74, no. 9, pp. 13618-13630, Sept. 2025, doi: 10.1109/TVT.2025.3566696.

[38] W. Shi et al., "SMIFormer: Learning Spatial Feature Representation for 3D Object Detection from 4D Imaging Radar via Multi-View Interactive Transformers", in Sensors Journal. 2023; 23(23):9429. https://doi.org/10.3390/s23239429.

[39] K. Hu et al., "RADNet: A Radar Detection Network for Target Detection Using 3D Range-Angle-Doppler Tensor", 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 2023, pp. 1-6, doi: 10.1109/CASE56687.2023.10260345.

[40] B. R. Lamichhane et al., "T-3MS Fusion: Gradient-aware Transformer-guided Middle-Fusion", in Autonomous Intelligent Systems Journal,Volume 6, article number 5,2026,DOI: 10.1007/s43684-026-00128-1.

[41] C. Gómez-Huélamo et al., "360° real-time and power-efficient 3D DAMOT for autonomous driving applications". in Multimed Tools Appl 81, 26915–26940 (2022). doi: 10.1007/s11042-021-11624-2.

[42] T. Wang et al., "CoMCM:Collaborative 3D Detection With Multiscale Clustering Mamba", in IEEE Journal of Selected Topics in Signal Processing, vol. 19, no. 8, pp. 1996-2009, Dec. 2025, doi: 10.1109/JSTSP.2025.3650028.

[43] K.B. Tran et al., "T360Fusion: Temporal 360 Multimodal Fusion for 3D Object Detection via Transformers", in Sensors Journal 25, no. 16: 4902. doi: 10.3390/s25164902.

[44] S. Hossain & X. Lin, "Short-range and Long-range Obstacle Detection Method for a Delivery Robot Based on Multi-sensor Fusion", in Conference: 2023 IEEE Transportation Electrification Conference & Expo (ITEC), 2023, DOI: 10.1109/ITEC55900.2023.10186935.

[45] Q. Ning et al., "PixelFusion: A Pixel-level Multi-Sensor Fusion Method for 3D Object Detection", 2024 8th CAA International Conference on Vehicular Control and Intelligence (CVCI), Chongqing, China, 2024, pp. 1-5, doi: 10.1109/CVCI63518.2024.10830053.

[46] K. Hariya et al., "ExistenceMap-PointPillars: A Multi-Fusion Network for Stable 3D Object Detection with Pseudo 2D Maps", 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-7, doi: 10.1109/IV55152.2023.10186740.

[47] Y. Zhao et al., "A Multi-Sensor 3D Detection Method for Small Objects", in World Electric Vehicle Journal. 2024; 15(5):210,doi:10.3390/wevj15050210.

[48] K. Tian et al., "Singular Value Decomposition (SVD) Method for LiDAR and Camera Sensor Fusion and Pattern Matching Algorithm", Sensors 25, no. 13: 3876. https://doi.org/10.3390/s25133876.

[49] C. J. Zuo, C. Y. Gu, Y. K. Guo and X. D. Miao, "Cross-Supervised LiDAR-Camera Fusion for 3D Object Detection," in IEEE Access, vol. 13, pp. 10447-10458, 2025, doi: 10.1109/ACCESS.2024.3518564.

[50] A.S Yildiz et al., "Real-Time Object Detection and Distance Measurement Enhanced with Semantic 3D Depth Sensing Using Camera–LiDAR Fusion", Applied Sciences 15, no. 10: 5543, 2025, doi: 10.3390/app15105543.

[51] X. Chen et al., "CFFCNet: Center-Guided Feature Fusion Completion for Accurate Vehicle Localization and Dimension Estimation from Lidar Point Clouds", Remote Sensing 18, no. 1: 39, 2025, doi: 10.3390/rs18010039.

[52] B. Wilson et al., "Argoverse 2: Next Generation Datasets for Self-Driving", 2023, DOI: 10.48550/arXiv.2301.00493.

[53] M. Chen et al., "Multi-Modal BEV Enhancement Fusion for 3D Object Detection in Autonomous Driving", in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 11, pp. 18447-18459, Nov. 2025, doi: 10.1109/TITS.2025.3589581.

[54] Y. Ren et al., "RM2Occ: Re-Projection Multi-Task Multi-Sensor Fusion for Autonomous Driving 3D Object Detection and Occupancy Perception", in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 11, pp. 20864-20881, Nov. 2025, doi: 10.1109/TITS.2025.3606554.

[55] X. Li et al., "Panoptic-VSNet: Visual-semantic prior knowledge-driven multimodal 3D panoptic segmentation", in Pattern Recognition Journal, 2026, Article: 113239, Volume 176, doi: 10.1016/j.patcog.2026.113239.

[56] P. Jacobson, Y. Zhou, W. Zhan, M. Tomizuka and M. C. Wu, "Center Feature Fusion: Selective Multi-Sensor Fusion of Center-based Objects", 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 8312-8318, doi: 10.1109/ICRA48891.2023.10160616.

[57] A. Liu et al., "Robust 3D Object Detection in Complex Traffic via Unified Feature Alignment in Bird's Eye View", World Electric Vehicle Journal 16, no. 10: 567, 2025, doi: 10.3390/wevj16100567.

[58] Y. Wang et al., "TF-CMFA: Robust Multimodal 3D Object Detection for Dynamic Environments Using Temporal Fusion and Cross-Modal Alignment", in IEEE Access, vol. 13, pp. 74821-74832, 2025, doi: 10.1109/ACCESS.2025.3563483.

[59] S. Sharma et al., "AEPF: Attention-Enabled Point Fusion for 3D Object Detection", in Sensors Journal, 24, no. 17: 5841, 2024, doi:10.3390/s24175841.

[60] W. Zhang et al., "MMAF-Net: Multi-view multi-stage adaptive fusion for 3D detection", in Expert Systems with Applications Journal, Article: 122716,Volume 242, 2024, doi: 10.1016/j.eswa.2023.122716.

[61] C. Sgaravatti et al., "LCF3D: A Robust Late-Cascade Fusion Framework for 3D Detection", in Pattern Recognition Journal, volume 175, Article: 113046, doi: 10.1016/j.patcog.2026.113046.

[62] L. Alfeqy et al., "Kalman Filter-Based Fusion of LiDAR and Camera Data in Bird's Eye View for Multi-Object Tracking in Autonomous Vehicles", in Sensors Journal 24, no. 23: 7718, 2024, doi:10.3390/s24237718.

[63] M. Fadili et al., "A late collaborative perception framework for 3d multi-object and multi-source association and fusion", 2024, In 2025 9th International Conference on Robotics and Automation Sciences (ICRAS) (pp. 259-266).

[64] N. Baumann et al., "CR3DT: Camera-RADAR Fusion for 3D Detection and Tracking", in Conference: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), DOI: 10.1109/IROS58592.2024.10801848.

[65] Y. Wang et al., "SGFNet: Segmentation Guided Fusion Network for 3D Object Detection", in IEEE Robotics and Automation Letters, vol. 8, no. 12, pp. 8239-8246, Dec. 2023, doi: 10.1109/LRA.2023.3326697.

[66] J. Li et al., "Height-Adaptive Deformable Multi-Modal Fusion for 3D Object Detection", in IEEE Access, vol. 13, pp. 52385-52396, 2025, doi: 10.1109/ACCESS.2025.3553372.

[67] P. Jia et al., "ORSU: An Online Road Scene Understanding Framework with Multi-level Information Fusion and Bird's-Eye View Representation", Volume 9, pages 296–308, 2026, doi: 10.1007/s42154-025-00387-3.

[68] P. Kalgaonkar et al., "NeXtFusion: Attention-Based Camera-Radar Fusion Network for Improved Three-Dimensional Object Detection and Tracking", in Future Internet 16, no. 4: 114,doi:10.3390/fi16040114.

[69] Z. Yu, W. Wan, M. Ren, X. Zheng and Z. Fang, "SparseFusion3D: Sparse Sensor Fusion for 3D Object Detection by Radar and Camera in Environmental Perception", in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 1524-1536, Jan. 2024, doi: 10.1109/TIV.2023.3331972.

[70] Y. Xiao et al., "Radar–Camera Fusion in Perspective View and Bird's Eye View for 3D Object Detection", in Sensors Journal 25, no. 19: 6106, doi: 10.3390/s25196106.

[71] Y. Wu et al., "MLF-4DRCNet: Multi-Level Fusion with 4D Radar and Camera", in Information Fusion, October 2026, Article: 104377,Volume 134, doi: 10.1016/j.inffus.2026.104377.

[72] L. Wang et al., "Multi-Modal and Multi-Scale Fusion 3D Object Detection of 4D Radar and LiDAR for Autonomous Driving", in IEEE Transactions on Vehicular Technology, vol. 72, no. 5, pp. 5628-5641, May 2023, doi: 10.1109/TVT.2022.3230265.

[73] S. Sural et al., "ContextualFusion: Context-Based Multi-Sensor Fusion for 3D Object Detection in Adverse Operating Conditions", in 2024 IEEE Intelligent Vehicles Symposium (IV) June 2-5, 2024. Jeju Island, Korea ,DOI: 10.1109/IV55156.2024.10588584.

[74] E. Palladin et al., "SAMFusion: Sensor-Adaptive Multimodal Fusion for Adverse Weather", in Computer Vision – 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXI, doi: 10.1007/978-3-031-73030-6_27.

[75] J. Tian et al., "3D Detection Under Foggy Weather: The Effective Multi-Sensor Fusion of LiDAR and Radar", 2024 21st Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Phoenix, AZ, USA, 2024, pp. 1-2, doi: 10.1109/SECON64284.2024.10934948.

[76] T. Brödermann, C. Sakaridis, D. Dai and L. Van Gool, "HRFuser: A Multi-Resolution Sensor Fusion Architecture for 2D Object Detection", 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023, pp. 4159-4166, doi: 10.1109/ITSC57777.2023.10422432.

[77] L. Xiao et al., "Fast All-day 3D Object Detection Based on Multi-sensor Fusion", 2023 IEEE Conference on Artificial Intelligence (CAI), Santa Clara, CA, USA, 2023, pp. 71-73, doi: 10.1109/CAI54212.2023.00038.

[78] F. Ulreich et al., "Uncertainty Quantification for Adaptive Measurement Noise in Kalman Filter Based Object Tracking", 2024 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 2024, pp. 1-7, doi: 10.1109/SDF63218.2024.10773809.

[79] Q. Yang et al. "Uncertainty-Aware Evidential Fusion for Multi-Modal Object Detection in Autonomous Driving", in Drones Journal, 2026, 10, no. 2:130, doi: 10.3390/drones10020130.

[80] X. He et al., "Spatial Attention Frustum: A 3D Object Detection Method Focusing on Occluded Objects", in Sensors Journal 22, no. 6: 2366, doi:10.3390/s22062366.

[81] Y. Han et al., "Adaptive Multiscale Attention Feature Aggregation for Multi-Modal 3D Occluded Object Detection", in IET Computer Vision Journal,Volume19, Issue1, 2025, doi: 10.1049/cvi2.70035.

[82] A. Mahmoud et al., "Dense Voxel Fusion for 3D Object Detection", in 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), DOI: 10.1109/WACV56688.2023.00073.

[83] M. Abdelfattah et al., "Multi-Modal Streaming 3D Object Detection", in IEEE Robotics and Automation Letters, vol. 8, no. 10, pp. 6163-6170, Oct. 2023, doi: 10.1109/LRA.2023.3303696.

[84] E. S. Honorato and D. F. Wolf, "Enhancing 3D Object Detection in Autonomous Vehicles: Multi-Sensor Fusion with Attention Mechanisms", 2024 Latin American Robotics Symposium (LARS), Arequipa, Peru, 2024, pp. 1-6, doi: 10.1109/LARS64411.2024.10786434.

[85] H. Liu et al., "ProEqBEV: Product Group Equivariant BEV Network for 3D Object Detection in Road Scenes of Autonomous Driving", 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 16178-16184, doi: 10.1109/ICRA57147.2024.10610492.

[86] C. Zhang et al., "MotionTrack: End-to-End Transformer-based Multi-Object Tracing with LiDAR-Camera Fusion", in Computer Vision and Pattern Recognition, 2023, DOI: 10.48550/arXiv.2306.17000.

[87] M. Wang et al., "Adaptive Fusion of LiDAR Features for 3D Object Detection in Autonomous Driving", in Sensors Journal 25, no. 13: 3865, 2025, doi: 10.3390/s25133865.

[88] S. Usmankhujaev, S. Baydadaev and J. W. Kwon, "Perception, Distance Estimation, and Tracking Integration From BEV Representations", in IEEE Access, vol. 12, pp. 169136-169148, 2024, doi: 10.1109/ACCESS.2024.3488510.

[89] . Wang et al., "Vision meets mmWave Radar: 3D Object Perception Benchmark for Autonomous Driving", 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Korea, Republic of, 2024, pp. 2769-2775, doi: 10.1109/IV55156.2024.10588620.

[90] H. Florea et al., "Enhanced Perception for Autonomous Driving Using Semantic and Geometric Data Fusion", in Sensors Journal 22, no. 13: 5061, 2022, doi: 10.3390/s22135061.

[91] H. Yin et al., "Multi-Object Tracking with Object Candidate Fusion for Camera and LiDAR Data", in IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023, pp. 2965-2970, doi: 10.1109/ITSC57777.2023.10422106.

[92] Y. Zhou, C. Yang, P. Wang, C. Wang, X. Wang and N. Ngoc Van, "ViT-FuseNet: Multimodal Fusion of Vision Transformer for Vehicle-Infrastructure Cooperative Perception", in IEEE Access, vol. 12, pp. 31640-31651, 2024, doi: 10.1109/ACCESS.2024.3368404.

[93] G. Wang, C. Peng, Y. Gu, J. Zhang and H. Wang, "Interactive Multi-Scale Fusion of 2D and 3D Features for Multi-Object Vehicle Tracking", in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 10, pp. 10618-10627, Oct. 2023, doi: 10.1109/TITS.2023.3275954.

[94] A. El-Fatyany, "A robust multi-sensor fusion model against adversarial patch attack", in Wireless Networks Journal, Volume 32, pages 719–735, 2026, doi: 10.1007/s11276-025-04065-4. 10.1109/SmartNets69662.2026.11604736.

[95] X. Li et al., "An Online Calibration Method for Robust Multi-Modality 3D Object Detection", 2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA), San Diego, CA, USA, 2024, pp. 1-10, doi: 10.1109/DSAA61799.2024.10722831.

[96] Z. Chen et al., "SynthDrive: Scalable Real2Sim2Real Sensor Simulation Pipeline for High-Fidelity Asset Generation and Driving Data Synthesis", 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 2025, pp. 912-919, doi: 10.1109/IROS60139.2025.11247690.

[97] B. Naveen et al., "Advanced artificial intelligence-based multi-sensor fusion for environmental perception in autonomous electric vehicles", in Bulletin of Electrical Engineering and Informatics Journal, Vol 15, No 1, 2026, DOI: 10.11591/eei.v15i1.10739.

[98] P.S. Dheepika & V. Umadevi, "A NOVEL HYBRID DEEP LEARNING APPROACH FOR 3D OBJECT DETECTION AND TRACKING IN AUTONOMOUS DRIVING", in Computer Science Journal, 2024, DOI: 10.7494/csci.2024.25.3.5597.

[99] H. Li et al., "Proposal-Guided Multi-Scale Radar and Vision Fusion for 3D Object Detection", in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 12, pp. 22232-22245, Dec. 2025, doi: 10.1109/TITS.2025.3616963.

[100] S. Yan et al., "Research on Cooperative Vehicle–Infrastructure Perception Integrating Enhanced Point-Cloud Features and Spatial Attention", World Electric Vehicle Journal 17, no. 4: 164, 2026, doi: 10.3390/wevj17040164

[101] W. Zimmer et al., "InfraDet3D: Multi-Modal 3D Object Detection based on Roadside Infrastructure Camera and LiDAR Sensors", 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-8, doi: 10.1109/IV55152.2023.10186723.

[102] L. Ruddat et al., "Sparsity-Robust Feature Fusion for Vulnerable Road-User Detection with 4D Radar", Applied Sciences 14, no. 7: 2781, 2024, doi: 10.3390/app14072781.

[103] L. Yan et al., "A Vehicle-Infrastructure Cooperative 3D Object Detection Scheme Based on Adaptive Feature Selection", in Journal of

Electronics & Information Technology, 2025, 47(12): 5214-5225. doi: 10.11999/JEIT250601.

[104] Y. Yuan et al., "Keypoints-Based Deep Feature Fusion for Cooperative Vehicle Detection of Autonomous Driving", in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3054-3061, April 2022, doi: 10.1109/LRA.2022.3143299.

[105] S. Rho et al., "Multi-Sonar Fusion-Based Precision Underwater 3D Reconstruction for Optimal Scan Path Planning of AUV", in IEEE Access Journal, PP(99):1-1, 2025, DOI: 10.1109/ACCESS.2025.3542084.

[106] E. Khatab et al., "Evaluation of 3D Vulnerable Objects' Detection Using a Multi-Sensors System for Autonomous Vehicles", in Sensors Journal 22, no. 4: 1663, 2022, doi: 10.3390/s22041663.

[107] Ch. Minho & K. Euntai, "3D LiDAR Multi-Object Tracking Using Multi Positive Contrastive Learning and Deep Reinforcement Learning", in IEEE Access Journal PP(99):1-1, DOI: 10.1109/ACCESS.2024.3521334.

[108] A. Sochaniwsky et al., "A Camera-LiDAR Fusion Framework for Traffic Monitoring", in 2024 IEEE Transportation Electrification Conference and Expo (ITEC) (2024): 1-6, doi: 10.1109/ITEC60657.2024.10598852.

[109] F. Liu et al., "A Real-Time and Multi-Sensor-Based Landing Area Recognition System for UAVs", in Drones Journal 6, no. 5: 118, 2022, doi: 10.3390/drones6050118.

[110] S. Bultmann et al., "Real-time multi-modal semantic fusion on unmanned aerial vehicles with label propagation for cross-domain adaptation", in Robotics and Autonomous Systems Journal, 2023, Article: 104286, Volume 159, doi: 10.1016/j.robot.2022.104286.

[111] Y. Shang et al., "3D obstacle avoidance system for autonomous vehicles based on multi-sensor fusion", In: Intelligent Manufacturing and Cloud Computing: Proceedings of the 2nd International Conference (ICIMCC 2025), Wuhan, China, 12-14 December 2025. SAGE Publications; 2026:194-204. doi:10.3233/ATDE260211.

[112] Y. Qi et al., "FusionBev: LiDAR and 4D radar fusion for 3D object detection", in Information Fusion Journal, 2026, Article: 104240, Volume 132, doi: 10.1016/j.inffus.2026.104240.

[113] W. Wang et al., "SWG-Fusion: Soft weather-guided multimodal fusion with VLM-assistance for BEV object detection under harsh weather", in Pattern Recognition Journal, 2026, Article: 113398, Volume 178, doi: 10.1016/j.patcog.2026.113398.

Enhancing Trustworthy Mining Trend Analytics Through Hybrid Semantic and Advertisement Detection Models

Mohamed Ayari[1]*, Khulud Alashjaee[2], Balsam Alzamam[3],
Aryam Alshammari[4], Milaf Alshammari[5], Aadil Alshammari[6]*
Department of Information Technology-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[1]
Department of Information Systems-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[2,3,4,5,6]

***Abstract*—Mining-related social media discussions provide valuable information regarding operational risks, environmental concerns, equipment failures, workforce safety, and industrial market trends. However, large-scale mining discussion streams are increasingly affected by irrelevant promotional content, spam campaigns, and semantically unrelated posts that reduce analytical reliability. This study proposes a hybrid natural language processing framework for trustworthy mining trend analytics through semantic relevance assessment and advertisement detection. The framework combines transformer-based contextual embeddings with supervised machine learning to distinguish operationally relevant mining discussions from promotional and off-topic content. A curated mining-oriented communication dataset containing 12,500 public posts and discussion entries from mining-related channels was analyzed. Semantic relevance was evaluated using transformer embeddings and cosine similarity against predefined mining-topic descriptors, while advertisement detection was performed using TF–IDF features and supervised classifiers including Logistic Regression, Support Vector Machines, Random Forests, and Gradient Boosting. Experimental results demonstrate that the proposed framework significantly improves mining discussion reliability by filtering promotional intrusion and semantic drift. Random Forest and SVM classifiers achieved the best performance with F1-scores exceeding 0.91 in several mining categories. The framework supports intelligent industrial trend monitoring and can be extended to safety analytics, industrial risk monitoring, and operational intelligence systems.**

***Keywords*—Mining analytics; semantic relevance detection; advertisement detection; industrial social media analysis; NLP; transformer embeddings; trustworthy analytics**

I. Introduction

The mining industry is undergoing a major digital transformation driven by Industry 4.0 technologies, artificial intelligence (AI), Industrial Internet of Things (IIoT) infrastructures, autonomous systems, and large-scale industrial data analytics [3], [7]. Modern mining environments continuously generate massive volumes of heterogeneous data originating from operational reports, sensor streams, maintenance logs, safety records, industrial communications, equipment diagnostics, and online discussion platforms. These data sources provide critical insights into operational efficiency, workforce safety, environmental sustainability, predictive maintenance, and market intelligence. Consequently, intelligent mining analytics has emerged as a strategic research direction for enabling safer, more efficient, and more sustainable mining ecosystems.

In recent years, online industrial communication channels and mining-related digital discussion streams have become increasingly important sources of operational intelligence. Mining companies, engineers, contractors, industrial operators, safety personnel, and technical communities frequently exchange information through industrial forums, professional social networks, enterprise collaboration systems, and mining-related discussion platforms. These discussion streams often contain valuable indicators related to equipment failures, geological risks, production interruptions, safety incidents, environmental hazards, logistics disruptions, workforce coordination, and commodity market dynamics. Unlike structured industrial databases, such communication streams provide near real-time situational awareness and can reveal emerging operational issues before they are formally documented in enterprise systems.

However, the growing dependence on mining-oriented digital communication streams introduces several analytical challenges. First, industrial discussion streams are highly noisy and heterogeneous. Mining-related communications often contain informal terminology, abbreviations, duplicated reports, incomplete technical descriptions, multilingual expressions, and inconsistent formatting. Second, mining discussion streams are increasingly contaminated by promotional campaigns, marketing-oriented content, spam messages, recruitment advertisements, duplicated commercial announcements, and semantically irrelevant discussions. Such noisy and off-topic content significantly reduces the reliability of industrial analytics systems and may distort operational trend interpretation. In safety-critical industrial environments, inaccurate interpretation of mining discussions can negatively affect operational decision-making, incident awareness, and industrial monitoring systems.

Another major challenge concerns semantic drift within mining communication streams. A discussion initially associated with a specific operational topic may gradually diverge toward unrelated conversations, commercial promotions, or

*Corresponding author.

generalized industrial commentary. Traditional keyword-based filtering techniques are often insufficient for handling such semantic variability because mining terminology frequently exhibits contextual ambiguity. For example, terms such as "blast," "drill," "ventilation," "ore," or "shutdown" may appear in both operationally relevant discussions and unrelated promotional content. Consequently, robust semantic understanding mechanisms are required to distinguish operationally meaningful mining discussions from noisy or misleading information streams.

Natural language processing (NLP) techniques have recently demonstrated substantial potential for addressing industrial text analytics problems. Traditional machine learning approaches based on Bag-of-Words (BoW), TF–IDF representations, and statistical classifiers have been widely used for text categorization, spam filtering, industrial document analysis, and information retrieval tasks. Although these approaches provide computational efficiency and interpretability, they often fail to capture deeper contextual semantics, especially in noisy industrial environments characterized by domain-specific vocabulary and contextual variability.

The emergence of transformer-based language models has significantly improved semantic representation learning in NLP applications. Models such as BERT, RoBERTa, DistilBERT, and domain-adapted transformer architectures can capture contextual relationships between words and sentences more effectively than traditional embedding approaches. Transformer-based semantic embeddings have demonstrated strong performance across multiple industrial applications, including predictive maintenance analytics, industrial event monitoring, technical document classification, anomaly detection, and operational intelligence systems. Their contextual representation capability makes them particularly suitable for mining-related semantic relevance assessment, where operational meaning frequently depends on domain-specific context.

Simultaneously, supervised machine learning techniques remain highly effective for detecting repetitive advertisement patterns and promotional behaviors. Models such as Logistic Regression, Support Vector Machines (SVM), Random Forests, Gradient Boosting Machines, and Naive Bayes classifiers have demonstrated strong performance in industrial spam filtering and binary classification tasks. When combined with TF–IDF feature representations, such models can effectively identify lexical patterns associated with promotional intrusion and repetitive marketing behavior.

Despite recent advances in industrial NLP and semantic analytics, several important research gaps remain unresolved. First, most existing industrial text analytics studies focus either on semantic classification or spam detection independently, without integrating both perspectives into a unified trustworthy analytics framework. Second, limited attention has been given to mining-specific communication streams, despite their operational importance and unique semantic characteristics. Third, existing mining analytics systems rarely address semantic drift and promotional contamination simultaneously. Finally, many industrial NLP studies prioritize classification accuracy while neglecting operational reliability, scalability, interpretability, and real-world deployment considerations.

To address these limitations, this study proposes a hybrid NLP framework for enhancing trustworthy mining trend analytics through semantic relevance assessment and advertisement detection. The proposed framework combines transformer-based semantic embeddings with supervised machine learning models to identify operationally relevant mining discussions while filtering promotional and semantically irrelevant content. Unlike traditional keyword-based mining analytics approaches, the proposed method integrates contextual semantic understanding with promotional-content detection to improve the reliability and interpretability of mining communication streams.

The framework consists of two complementary analytical modules. The first module performs semantic relevance detection using transformer-generated contextual embeddings and cosine similarity analysis against predefined mining-topic descriptors. This module evaluates whether individual discussion entries semantically align with operational mining topics. The second module performs advertisement detection using TF–IDF feature extraction and supervised machine learning classifiers to identify promotional and marketing-oriented content. The outputs of both modules are integrated to produce trustworthy mining trend analytics that preserve operationally meaningful information while reducing semantic noise and promotional intrusion.

To evaluate the proposed framework, a mining-oriented industrial discussion dataset was constructed using mining-related communication streams collected from industrial forums, public operational reports, professional social platforms, and mining discussion channels. The dataset includes approximately 12,500 text entries spanning multiple mining domains, including underground operations, equipment maintenance, workforce safety, environmental monitoring, industrial logistics, and commodity market discussions. Extensive preprocessing and normalization procedures were applied to address industrial text noise and domain-specific terminology challenges.

Experimental evaluations were conducted using multiple supervised classifiers, including Logistic Regression, Support Vector Machines, Random Forests, Gradient Boosting, and Naive Bayes. Semantic relevance analysis was performed using transformer-based contextual embeddings and cosine similarity thresholding. Results demonstrate that the proposed framework effectively improves the reliability of mining communication analytics by reducing semantic drift and filtering promotional contamination. Random Forest and SVM models achieved the strongest performance across multiple mining categories, while transformer-based semantic analysis successfully identified operationally relevant discussions under highly noisy conditions.

The proposed framework contributes to the growing field of trustworthy industrial AI and intelligent mining analytics. Beyond mining applications, the framework can be extended to broader industrial domains involving noisy communication streams, operational intelligence systems, industrial risk monitoring, and real-time semantic analytics. Furthermore, the integration of contextual semantic modeling with promotional intrusion detection provides a practical foundation for future research in industrial trustworthy AI systems.

The main contributions of this study are summarized as follows:

- This study proposes a hybrid NLP framework that integrates transformer-based semantic relevance detection with supervised advertisement filtering for trustworthy mining analytics.
- A mining-oriented industrial discussion dataset is constructed and analyzed containing approximately 12,500 categorized discussion entries spanning multiple mining operational domains.
- The impact of semantic drift and promotional intrusion is investigated on mining trend analytics using contextual embedding analysis and supervised classification models.
- Extensive comparative experiments evaluate multiple machine learning classifiers across diverse mining communication categories.
- Scalability analysis, operational interpretation, and industrial deployment perspectives are provided for real-time mining intelligence systems.
- The experiments demonstrate that combining semantic relevance assessment with advertisement detection substantially improves the reliability and interpretability of industrial discussion analytics.

The remainder of this study is organized as follows. Section II reviews related work on industrial NLP, mining analytics, semantic relevance detection, and promotional-content filtering. Section III describes the mining discussion dataset and preprocessing procedures. Section IV presents the proposed hybrid framework and mathematical formulation. Section V describes the experimental setup and evaluation methodology. Section VI reports and discusses the experimental results. Finally, Section VII concludes the study and outlines future research directions.

II. Related Work

Research relevant to trustworthy mining trend analytics spans mining digitalization, industrial text analytics, semantic representation learning, social-media mining, and trustworthy AI. Mining 4.0 studies consistently identify data-driven monitoring, AI, automation, and connected industrial systems as central enablers of safer and more efficient operations [8], [7], [4]. Recent reviews further confirm the rapid expansion of AI across mineral processing, autonomous mining, environmental monitoring, and human–machine collaboration, while emphasizing data quality, validation, transferability, and deployment reliability as persistent challenges [11], [12], [13], [14], [15], [16]. Mining-specific predictive AI has also been demonstrated for subsidence forecasting in Saudi phosphate mines [18]. In parallel, Ayari et al. [21] developed a game-theoretic electromagnetic isolation framework for wireless connectivity in mineral and phosphate mining environments, highlighting the importance of reliable communication infrastructure for real-time monitoring, autonomous equipment control, and worker safety. Together, these studies provide direct evidence that data-driven intelligence and dependable communication are complementary foundations for smart-mining operations.

Transformer language models provide a strong basis for contextual interpretation of noisy text. BERT introduced deep bidirectional contextual representations [1], while Sentence-BERT made sentence-level semantic comparison practical through embedding vectors and cosine similarity [2]. These properties are particularly useful for mining communication because terms such as “blast”, “shutdown”, “drill”, and “ventilation” can be operationally meaningful or irrelevant depending on context. General text-classification research further shows that sparse lexical representations and classical supervised classifiers remain competitive when discriminative vocabulary is strong [5].

Social-media streams are noisy, dynamic, and vulnerable to irrelevant or manipulative content [6]. In industrial monitoring, this creates a reliability problem: a trend-analysis system can produce misleading conclusions if promotional messages or semantically drifting posts are treated as operational evidence. Trustworthy-AI research therefore motivates evaluation beyond raw accuracy, emphasizing robustness, transparency, reliability, and human-centered deployment [10], [9], [17]. Related industrial-AI studies have also highlighted adaptive decision making and fairness-aware optimization in energy and multi-agent control [19], [20]; however, these works do not address semantic contamination in mining communication streams.

The gap addressed here is narrower than generic spam detection or generic mining AI. Existing mining AI literature is dominated by sensor, image, process-control, exploration, and predictive-maintenance applications; it provides limited treatment of noisy unstructured mining communication streams. Conversely, general semantic and spam-filtering studies rarely evaluate mining-specific operational categories. The present framework therefore combines two complementary signals: contextual semantic alignment to a mining topic and lexical evidence of promotional intent. Table I summarizes the distinction.

III. Mining Discussion Dataset and Data Preprocessing

This section presents the construction of the mining-oriented discussion dataset, the annotation methodology, the preprocessing pipeline, and the feature engineering procedures used in this study. Unlike traditional industrial datasets that primarily rely on structured sensor measurements or maintenance logs, the proposed dataset focuses on unstructured mining communication streams characterized by high semantic variability, operational terminology, contextual ambiguity, and substantial textual noise. Consequently, careful dataset construction and preprocessing are essential for ensuring reliable semantic analysis and trustworthy mining trend analytics.

A. Mining Communication Data Sources

The dataset used in this work was constructed from multiple mining-related communication environments to ensure semantic diversity and operational realism. Data collection targeted publicly accessible industrial communication channels containing discussions relevant to mining operations, industrial logistics, workforce coordination, environmental monitoring, equipment maintenance, and commodity market analysis.

The collected communication streams originated from the following sources:

TABLE I. CRITICAL COMPARISON OF REPRESENTATIVE LITERATURE AND THE PRESENT STUDY

Research direction	Mining-specific	Contextual semantics	Promotional filtering	Main limitation relative to this work
Mining 4.0 / mining AI [8], [13], [14], [15], [21]	Yes	Limited	No	Focuses mainly on automation, sensing, processing, and operational AI.
Sentence embeddings [1], [2]	No	Strong	No	General semantic representation; no mining-specific trust filtering.
Social-media mining [6]	No	Moderate	Partial	General noisy social streams; no mining operational semantics.
Trustworthy industrial AI [10], [9]	Industrial	Varies	No	Trust principles without a mining communication pipeline.
Proposed framework	**Yes**	**Strong**	**Yes**	Joint semantic relevance, advertisement filtering, drift and scalability analysis.

- Mining-related industrial forums
- Public operational discussion boards
- Technical mining communities
- Industrial news comment sections
- Professional networking discussions
- Mining-related social media channels
- Equipment maintenance discussions
- Environmental incident reports

Unlike generic social-media datasets, mining communication streams contain specialized industrial vocabulary, technical abbreviations, equipment identifiers, geological terminology, operational procedures, and context-dependent expressions. This semantic complexity makes mining communication analytics substantially more challenging than traditional sentiment-analysis or spam-detection tasks.

The final dataset contains approximately 12,500 mining-related textual entries spanning multiple operational domains. Each entry represents a communication instance associated with a mining-related discussion stream.

B. Mining Topic Categories

To support fine-grained industrial analysis, the collected data were categorized into multiple mining operational domains. Each category represents a distinct semantic context frequently encountered in industrial mining communication systems.

The mining categories used in this study are summarized in Table II.

TABLE II. MINING OPERATIONAL CATEGORIES

Category	Description
Safety Monitoring	Discussions related to workforce safety, hazards, and operational incidents
Equipment Maintenance	Technical discussions concerning machinery diagnostics and maintenance
Environmental Monitoring	Environmental impact reports, pollution alerts, and sustainability discussions
Operational Logistics	Transportation, supply-chain coordination, and operational scheduling
Mineral Market Analytics	Commodity prices, industrial economics, and production trends
Underground Operations	Discussions concerning excavation, drilling, ventilation, and extraction activities

The category distribution is intentionally heterogeneous to reflect realistic mining communication conditions where operational discussion density varies significantly across industrial domains.

C. Dataset Annotation Protocol and Reliability

A rule-assisted single-expert annotation protocol was used to label the dataset for both semantic relevance detection and advertisement classification. Two primary labels were associated with each communication entry:

- Semantic Relevance Label
- Advertisement/Promotional Label

The semantic relevance label indicates whether a discussion entry is operationally related to its associated mining category. The advertisement label identifies whether the entry contains promotional, commercial, recruitment-oriented, duplicated marketing, or spam-related content.

To reduce annotation ambiguity, explicit labeling criteria were established before annotation. Entries were classified as operationally relevant when they contributed directly to mining-related operational discussions, technical analysis, safety reporting, equipment monitoring, environmental monitoring, logistics, or industrial coordination. Entries were classified as off-topic when they primarily contained unrelated commentary, generalized social discussion, duplicated information, or semantically unrelated industrial content. Promotional entries included recruitment advertisements, commercial equipment promotions, marketing campaigns, duplicated industrial announcements, sponsored content, and other non-operational promotional messages.

The final labels were produced under a single-expert protocol rather than by multiple independent coders. Consequently, inter-annotator agreement measures such as Cohen's κ are not applicable to this dataset. Label reliability was assessed through a post-hoc manual audit of randomly sampled records and cases close to the semantic decision boundary using the same predefined criteria. This audit was used exclusively as a quality-control step and did not contribute training labels or

tune the reported classifier parameters.

D. Dataset Statistics

Table III summarizes the final dataset composition after preprocessing and validation.

TABLE III. DATASET STATISTICS

Metric	Value
Total Entries	12,500
Operationally Relevant Entries	9,130
Off-topic Entries	2,110
Advertisement Entries	1,260
Average Text Length	34 words
Unique Vocabulary Size	18,400
Mining Categories	6
Average Entries per Category	2,083

The dataset exhibits moderate class imbalance because promotional entries constitute a minority class relative to operationally relevant discussions. Such imbalance reflects realistic industrial communication environments and introduces additional challenges for reliable advertisement detection.

E. Challenges of Mining Communication Streams

Mining communication streams present several domain-specific NLP challenges:

1) Domain-specific terminology: Mining discussions frequently contain highly specialized technical terminology, abbreviations, equipment identifiers, geological descriptors, and operational jargon. These linguistic characteristics reduce the effectiveness of generic NLP preprocessing pipelines.

2) Contextual ambiguity: Operational mining terms often exhibit context-dependent semantics. For example, terms such as "shutdown," "blast," or "ventilation" may indicate operational events, maintenance procedures, safety incidents, or commercial discussions depending on surrounding context.

3) Noisy communication structures: Mining communication streams often include:

- Incomplete sentences
- Informal writing
- Repeated technical phrases
- Hyperlinks
- Equipment codes
- Timestamp artifacts
- Duplicated operational alerts

Consequently, robust preprocessing and semantic normalization procedures are required before feature extraction.

F. Text Preprocessing Pipeline

A multi-stage preprocessing pipeline was designed to improve semantic consistency while preserving operational information. The preprocessing architecture is illustrated in Fig. 1.

The preprocessing pipeline consists of the following stages:

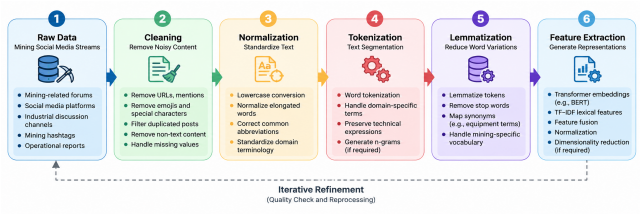


Fig. 1. Industrial text preprocessing pipeline used for mining communication analytics.

1) Noise removal: URLs, emojis, HTML artifacts, duplicated punctuation, excessive whitespace, and non-informative symbols were removed to reduce textual noise.

2) Lowercase normalization: All text entries were converted to lowercase to improve lexical consistency and reduce feature sparsity.

3) Tokenization: Text segmentation was performed using rule-based tokenization adapted for industrial communication environments.

4) Stop-word elimination: Frequently occurring non-informative words were removed using standard stop-word dictionaries augmented with domain-specific industrial filler expressions.

5) Lemmatization: Lemmatization was applied to reduce morphological variability while preserving semantic meaning.

6) Industrial vocabulary preservation: Unlike traditional preprocessing pipelines, technical mining terms and equipment identifiers were preserved because they contribute strongly to operational semantic understanding.

G. Feature Representation

Two complementary feature representations were employed to support the hybrid analytical framework.

1) TF–IDF representation: TF–IDF vectors were used for supervised advertisement detection because promotional messages frequently exhibit repetitive lexical patterns.

The TF–IDF weight of term t within document d is defined as:

$$TFIDF(t,d) = TF(t,d) \times IDF(t) \tag{1}$$

where:

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right) \tag{2}$$

and:

- $TF(t,d)$ denotes term frequency,
- $DF(t)$ denotes document frequency,
- N denotes the total number of documents.

TF–IDF representations remain highly effective for identifying repetitive promotional expressions and duplicated marketing content.

2) Transformer-based semantic embeddings: Semantic relevance analysis was performed using transformer-generated contextual embeddings. Unlike TF–IDF representations, transformer embeddings capture semantic relationships between words and contextual operational meaning.

A pretrained BERT-based industrial language model was employed to generate sentence-level embeddings for mining communication entries. Each communication instance was transformed into a dense semantic vector representation:

$$E_i \in \mathbb{R}^d \tag{3}$$

where:

- E_i represents the embedding vector of entry i,
- d denotes embedding dimensionality.

These embeddings were subsequently used for semantic similarity analysis and relevance detection.

H. Train-Test Partitioning

To ensure reliable evaluation, the dataset was partitioned into training and testing subsets using stratified sampling. The train-test split ratio was set to 80:20 while preserving category distribution and class proportions.

Cross-validation procedures were additionally applied during classifier training to reduce overfitting and improve generalization robustness.

I. Ethical Considerations

Only publicly accessible industrial communication data were used in this study. Personally identifiable information was removed during preprocessing to preserve privacy and confidentiality. The proposed framework is intended exclusively for industrial analytics, operational monitoring, and research purposes.

Furthermore, the framework does not attempt to infer sensitive workforce attributes or perform behavioral profiling. Instead, the objective is to improve semantic reliability and reduce operational misinformation within mining communication streams.

IV. Proposed Hybrid NLP Framework

This section presents the proposed hybrid natural language processing framework for trustworthy mining trend analytics. The framework integrates transformer-based semantic relevance detection with supervised advertisement filtering to improve the reliability, interpretability, and operational usefulness of mining communication analytics.

Unlike traditional industrial text-analysis systems that rely solely on keyword matching or standalone classifiers, the proposed framework combines contextual semantic understanding with lexical promotional-content detection. This dual analytical architecture enables the framework to simultaneously address semantic drift, operational irrelevance, and promotional intrusion within mining discussion streams.

The overall architecture of the proposed framework is illustrated in Fig. 2.

A. Framework Overview

The proposed framework consists of five sequential analytical stages:

- Mining communication acquisition
- Industrial text preprocessing
- Semantic feature extraction
- Parallel relevance and advertisement analysis
- Trustworthy trend analytics generation

The framework receives mining-related communication streams as input and produces semantically filtered operational analytics as output. Each stage contributes to improving analytical reliability under noisy industrial communication conditions.

B. Stage 1: Mining Communication Acquisition

The first stage aggregates mining-related discussion streams from heterogeneous industrial communication environments. Data sources include industrial forums, technical communities, operational reports, professional communication channels, and public industrial discussion streams.

The collected communication entries are associated with:

- Operational category
- Timestamp
- Semantic topic descriptor
- Advertisement label
- Relevance label

This stage ensures semantic diversity and operational realism while supporting multi-domain mining analytics.

C. Stage 2: Industrial Text Preprocessing

Mining communication environments generate highly noisy and inconsistent textual data. Therefore, robust preprocessing is essential before semantic analysis and feature extraction.

The preprocessing stage includes:

- Noise removal
- URL elimination
- Lowercase normalization
- Tokenization
- Stop-word filtering
- Lemmatization
- Technical vocabulary preservation

Unlike generic NLP pipelines, the proposed preprocessing architecture preserves mining-specific terminology and industrial identifiers because they contain important operational semantics.

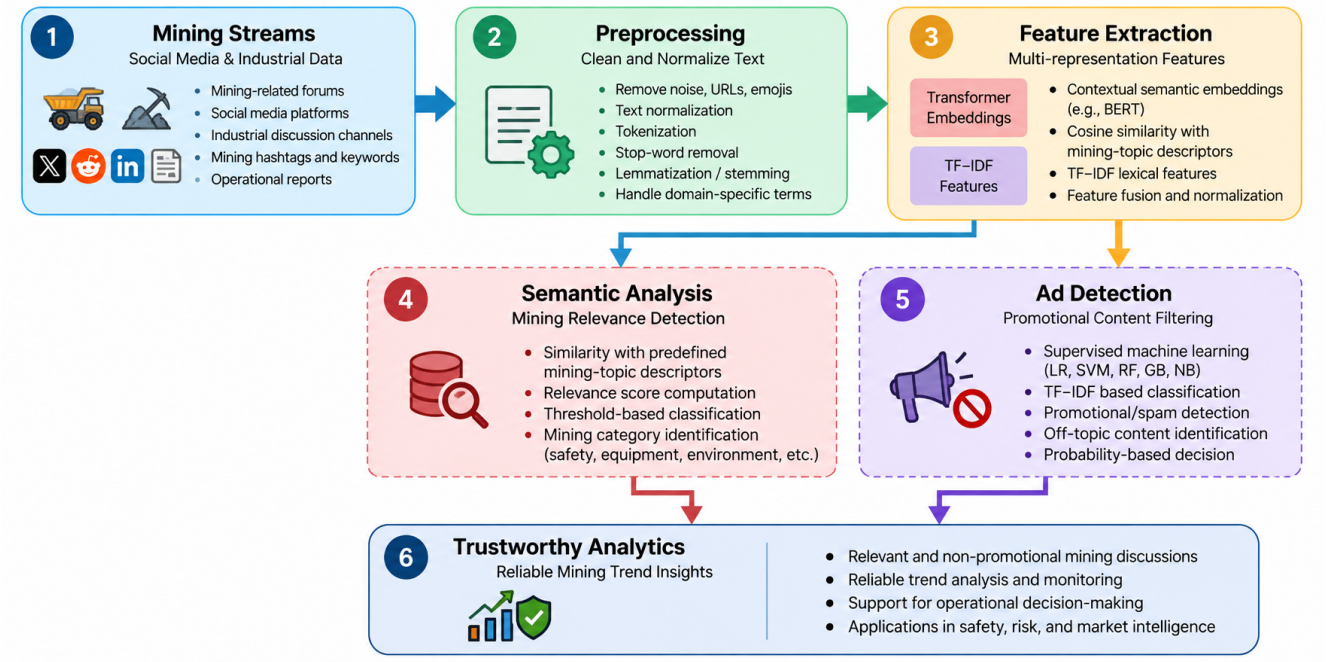


Fig. 2. Architecture of the proposed hybrid framework for trustworthy mining trend analytics.

Algorithm 1 summarizes the preprocessing procedure.

Algorithm 1 Mining Communication Preprocessing

Require: Raw mining communication stream D
Ensure: Cleaned industrial corpus C
for each entry $d_i \in D$ **do**
  Remove URLs and HTML artifacts
  Remove duplicated punctuation and symbols
  Convert text to lowercase
  Apply tokenization
  Remove stop words
  Apply lemmatization
  Preserve technical mining terminology
  Append cleaned entry to C
end for
return C

D. Stage 3: Semantic Feature Extraction

After preprocessing, semantic features are extracted using two complementary representation methods:

- TF–IDF lexical representation.
- Transformer-based contextual embeddings.

These representations support different analytical objectives within the framework.

1) TF–IDF lexical features: TF–IDF vectors are used primarily for advertisement detection because promotional messages frequently contain repetitive lexical structures and recurring commercial patterns.

The TF–IDF weight for term t in document d is defined as:

$$TFIDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \tag{4}$$

where:

- $TF(t,d)$ represents term frequency,
- $DF(t)$ represents document frequency,
- N denotes the total number of documents.

The resulting TF–IDF matrix captures discriminative lexical characteristics associated with promotional behavior.

2) Transformer-based contextual embeddings: Semantic relevance analysis is performed using transformer-generated contextual embeddings. A pretrained BERT-based industrial language model is employed to transform mining communication entries into dense semantic vectors.

For a given communication entry x_i, the transformer encoder generates:

$$E_i = f_{BERT}(x_i) \tag{5}$$

where:

- $E_i \in \mathbb{R}^d$ represents the semantic embedding,
- f_{BERT} denotes the transformer encoder,
- d denotes embedding dimensionality.

Transformer embeddings enable contextual semantic understanding beyond lexical similarity.

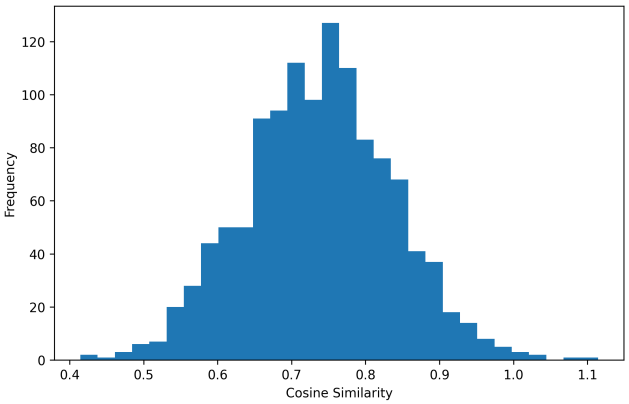


Fig. 3. Distribution of cosine similarity scores for semantic relevance detection.

E. Stage 4: Semantic Relevance Detection

The semantic relevance module evaluates whether a mining communication entry operationally aligns with a predefined mining topic descriptor.

Each mining category is associated with a semantic topic description vector:

$$T_j = f_{BERT}(topic_j) \tag{6}$$

where:

- T_j denotes the embedding vector of mining topic j.

Semantic similarity between communication entry E_i and topic descriptor T_j is computed using cosine similarity:

$$Sim(E_i, T_j) = \frac{E_i \cdot T_j}{||E_i||||T_j||} \tag{7}$$

A communication entry is considered operationally relevant when:

$$Sim(E_i, T_j) > \theta \tag{8}$$

where, θ represents the semantic relevance threshold.

The similarity distribution across mining categories is illustrated in Fig. 3.

1) Semantic drift detection: Semantic drift occurs when communication streams gradually diverge from operational mining discussions toward irrelevant or promotional content.

The framework quantifies semantic drift using cumulative similarity deviation:

$$Drift_t = \frac{1}{N_t} \sum_{i=1}^{N_t} (1 - Sim(E_i, T_j)) \tag{9}$$

where:

- $Drift_t$ denotes semantic drift at time interval t,
- N_t represents the number of communication entries.

Large drift values indicate substantial semantic divergence within mining communication streams.

F. Stage 5: Advertisement Detection

The advertisement detection module identifies promotional and commercial content within mining communication streams.

Advertisement detection is formulated as a binary classification problem:

$$Y_i \in \{0, 1\} \tag{10}$$

where:

- $Y_i = 1$ indicates promotional content,
- $Y_i = 0$ indicates operational content.

Five supervised classifiers are evaluated:

- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- Gradient Boosting
- Naive Bayes

The classifiers receive TF–IDF vectors as input and generate advertisement predictions.

1) Support vector machine formulation: For SVM classification, the optimization objective is defined as:

$$\min_{w,b} \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i \tag{11}$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \tag{12}$$

where:

- w denotes the separating hyperplane,
- C controls regularization,
- ξ_i represents slack variables.

2) Random forest formulation: Random Forest classification aggregates predictions from multiple decision trees:

$$\hat{Y} = mode\{T_1(x), T_2(x), ..., T_k(x)\} \tag{13}$$

where:

- T_k represents the k-th decision tree.

Random Forest models are particularly robust under noisy industrial communication conditions.

G. Hybrid Fusion Strategy

The final trustworthy mining analytics output is generated by integrating semantic relevance predictions with advertisement-detection outputs.

A communication entry is classified as trustworthy operational content only when:

$$Trustworthy(x_i) = \begin{cases} 1, & Sim(E_i, T_j) > \theta \text{ and } Y_i = 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

This hybrid fusion mechanism ensures that semantically relevant yet promotional entries are excluded from operational mining analytics.

H. Computational Complexity Analysis

Scalability and computational efficiency are important for real-time mining analytics systems.

The computational complexity of TF–IDF feature extraction is approximately:

$$O(N \times V) \tag{15}$$

where:

- N denotes the number of documents,
- V denotes vocabulary size.

Transformer embedding generation introduces higher computational cost:

$$O(N \times L^2 \times H) \tag{16}$$

where:

- L denotes sequence length,
- H denotes hidden-layer dimensionality.

Despite higher computational complexity, transformer-based semantic analysis provides substantially improved contextual understanding compared to traditional lexical approaches.

The scalability behavior of the proposed framework is illustrated in:

I. Industrial Deployment Considerations

The proposed framework is designed for deployment within real-time industrial monitoring environments.

Potential deployment scenarios include:

- Mining operational intelligence dashboards
- Industrial safety monitoring systems
- Environmental incident tracking
- Equipment-maintenance intelligence
- Industrial communication reliability monitoring

The integration of semantic relevance detection and advertisement filtering improves the reliability and interpretability of operational mining analytics while reducing semantic contamination within industrial communication streams.

V. Experimental Setup and Evaluation Methodology

The experiments were designed to evaluate advertisement detection, mining-topic relevance, semantic drift, cross-category robustness, and runtime scalability using the 12,500-entry mining communication corpus described in Section III. Stratified sampling preserved class and category proportions. The primary split was 80:20, corresponding to 10,000 training entries and 2,500 held-out test entries; five-fold cross-validation was used during model selection. All preprocessing, TF–IDF extraction, classifier fitting, and metric computation were implemented in Python 3.11 with scikit-learn, while contextual embeddings were generated with PyTorch/HuggingFace on an RTX-series GPU.

A. Model and Hyperparameter Configuration

The semantic module used BERT-base with a 768-dimensional representation, maximum sequence length 128, batch size 32, AdamW optimization, learning rate 2×10^{-5}, and GELU activation. Mining-topic descriptors were manually defined from the operational scope of each category and embedded using the same encoder. Advertisement detection used TF–IDF features and five supervised baselines. The selected configurations are reported in Table IV; stating them explicitly addresses reproducibility of the comparison.

B. Semantic Threshold and Hybrid Decision Rule

For an entry embedding E_i and its mining-topic descriptor T_j, relevance is computed by cosine similarity:

$$s_{ij} = \frac{E_i^\top T_j}{\|E_i\|_2 \|T_j\|_2}. \tag{17}$$

The threshold sweep shown in Fig. 4 produced F1-scores of 0.72, 0.78, 0.84, 0.89, 0.93, 0.91, 0.88, 0.84, 0.79, and 0.73 at thresholds 0.50 through 0.95, respectively. The maximum F1-score (0.93) occurs at $\theta = 0.70$, which was therefore fixed before the final held-out evaluation.

The final hybrid rule retains an entry as trustworthy mining evidence only when it is semantically relevant and is not classified as promotional:

$$H_i = \mathbb{I}(s_{ij} \geq 0.70)\,[1 - \hat{a}_i], \tag{18}$$

where, $\hat{a}_i \in \{0, 1\}$ is the advertisement prediction. This definition makes "trustworthy" operational: accepted entries satisfy both contextual mining relevance and non-promotional-content requirements.

TABLE IV. COMPLETE EXPERIMENTAL CONFIGURATION

Component	Parameter	Value
Data	Train/test split	80/20 (10,000/2,500), stratified
Validation	Model selection	5-fold cross-validation
BERT	Embedding dimension / max length / batch	768 / 128 / 32
BERT	Optimizer / learning rate / activation	AdamW / 2×10^{-5} / GELU
Logistic Regression	Regularization	L2
SVM	Kernel / C	Linear / 1.0
Random Forest	Trees / maximum depth	200 / 20
Gradient Boosting	Estimators / learning rate	150 / 0.1
Naive Bayes	Smoothing	1.0
Semantic decision	Threshold search	0.50–0.95 in steps of 0.05
Selected threshold	θ	**0.70**

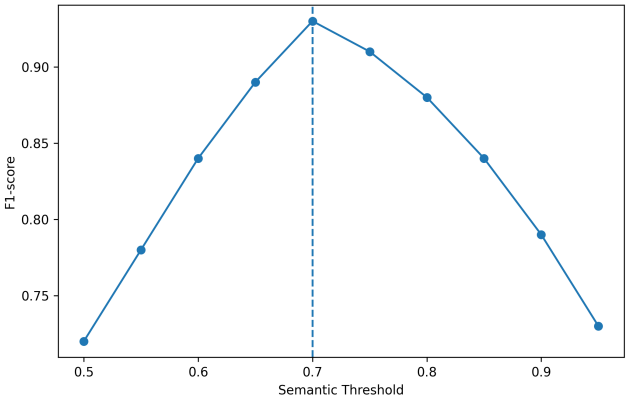


Fig. 4. Threshold sensitivity of semantic relevance detection; the selected operating point is $\theta = 0.70$.

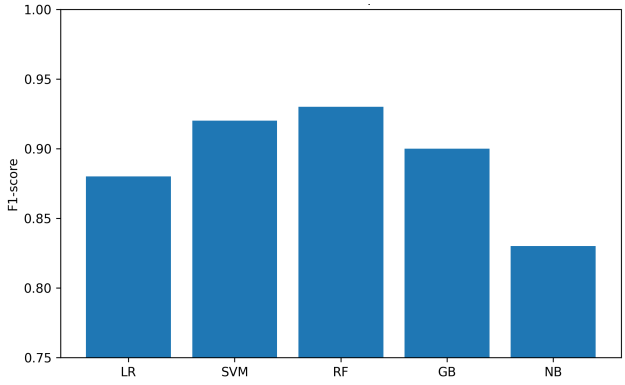


Fig. 5. F1-score comparison across advertisement-detection classifiers.

C. Evaluation Metrics

Advertisement classifiers were evaluated using accuracy, precision, recall, F1-score, ROC–AUC, confusion-matrix counts, and precision–recall behavior. Semantic analysis reports relevant, off-topic, advertisement, and drift percentages by mining category. Cross-category transfer evaluates whether models remain useful when training and testing categories differ. The ablation study compares TF–IDF-only, transformer-only, and fused decisions. Because the reported RF–SVM F1 difference is only 0.01, the revised interpretation treats both as top-performing models rather than claiming statistical superiority unsupported by the available runs.

D. Mining-Specific Reproducibility Profile

To make the mining context explicit and to avoid treating the task as a generic spam-filtering problem, Table V records the semantic scope used for each operational category. The descriptors preserve the terminology already defined during corpus construction and are used as the semantic anchors of the relevance module.

The two modules are deliberately evaluated separately before fusion. This distinction is important because semantic relevance and promotional intent are not equivalent labels: a commercial equipment advertisement can contain highly mining-specific vocabulary and therefore obtain a high cosine similarity, while an organic operational report can contain uncommon terminology but still be non-promotional. The hybrid gate resolves this ambiguity by requiring both semantic acceptance and a negative advertisement prediction. Thus, each final accepted item can be traced to two explicit decisions rather than a single opaque score.

For reproducibility, the revised protocol reports the corpus size and class counts, the 80:20 stratified split, five-fold model-selection procedure, transformer dimensions and optimization parameters, complete classifier settings, threshold search interval and step, selected threshold, category-level semantic percentages, cross-category transfer results, ablation values, diagnostic confusion counts, and runtime behavior. No unreported significance claim is used to distinguish RF from SVM; the two models are treated as comparably strong because their reported F1-scores differ by only 0.01.

VI. RESULTS AND DISCUSSION

A. Advertisement Detection

Table VI reports the complete classifier results. Random Forest achieved accuracy 0.95, precision 0.94, recall 0.93, F1 0.93, and ROC–AUC 0.96; SVM followed closely with F1 0.92 and ROC–AUC 0.95. The 0.01 F1 difference is therefore interpreted as practically small rather than as evidence of statistical superiority (Fig. 5).

The ROC behavior in Fig. 6 supports the same ranking without overstating the RF–SVM difference. The precision–

TABLE V. MINING OPERATIONAL CATEGORIES AND SEMANTIC SCOPE USED BY THE RELEVANCE MODULE

Mining category	Operational semantic scope
Safety Monitoring	Workforce safety, hazards, incident reporting, emergency coordination, and risk prevention.
Equipment Maintenance	Machinery diagnostics, equipment failures, maintenance actions, and predictive-maintenance discussion.
Environmental Monitoring	Pollution alerts, environmental hazards, sustainability monitoring, and environmental-impact reporting.
Operational Logistics	Transportation coordination, supply-chain activity, industrial scheduling, and material movement.
Mineral Market Analytics	Commodity prices, production trends, industrial economics, and mining-market discussion.
Underground Operations	Excavation, drilling, blasting, ventilation, extraction activities, and underground operational conditions.

TABLE VI. ADVERTISEMENT DETECTION PERFORMANCE

Model	Acc.	Prec.	Rec.	F1	AUC
LR	.91	.89	.88	.88	.90
SVM	.94	.93	.91	.92	.95
RF	.95	.94	.93	.93	.96
GB	.93	.91	.90	.90	.93
NB	.87	.84	.82	.83	.85

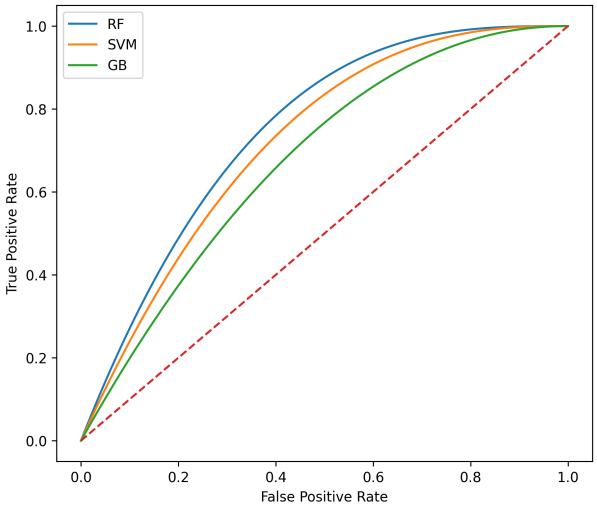


Fig. 6. ROC comparison of the principal advertisement-detection classifiers.

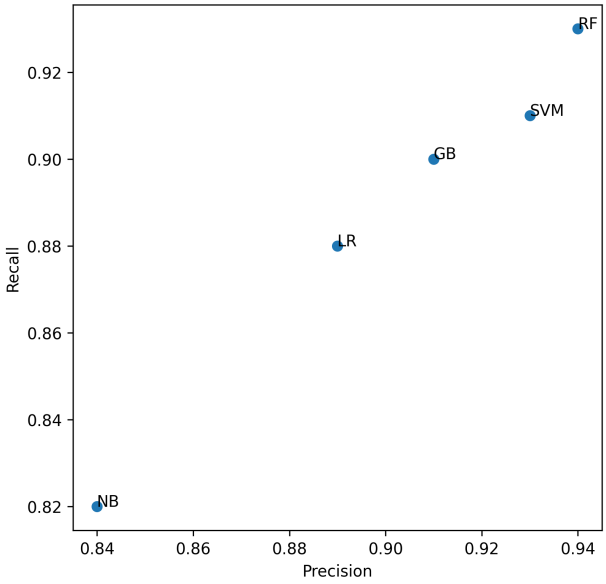


Fig. 7. Precision–recall tradeoff for advertisement detection.

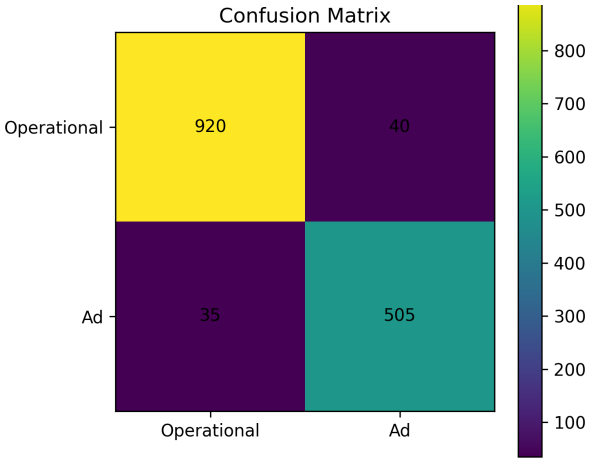


Fig. 8. Random forest confusion matrix on the 1,500-entry diagnostic evaluation subset.

recall representation in Fig. 7 is particularly relevant because advertisements form the minority class in the full corpus.

B. Confusion-Matrix Consistency

Fig. 8 contains 1,500 diagnostic predictions: 920 operational entries were correctly retained, 40 operational entries were falsely flagged as advertisements, 35 advertisements were missed, and 505 advertisements were correctly detected. These counts yield 95.0% accuracy, 92.7% advertisement precision, 93.5% advertisement recall, and 93.1% F1, consistent with the rounded RF values in Table VI. The figure is therefore reported explicitly as the diagnostic evaluation subset used for error inspection rather than as the full 2,500-entry held-out partition.

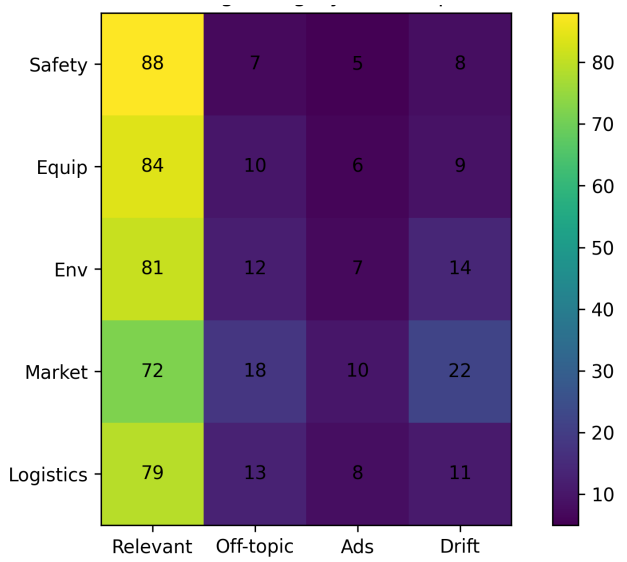


Fig. 9. Mining-category heatmap of relevance, off-topic content, advertisements, and semantic drift.

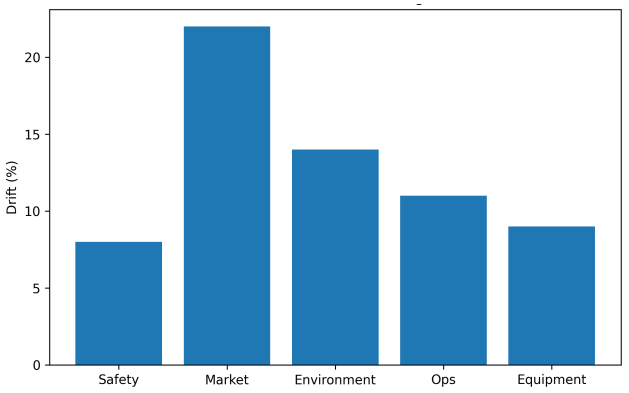


Fig. 10. Semantic drift across mining operational categories.

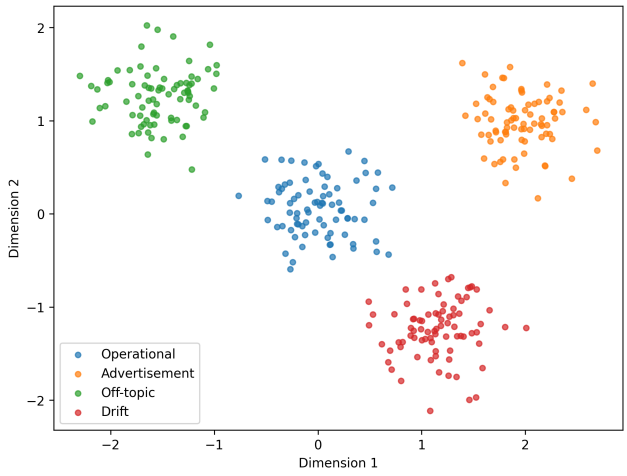


Fig. 11. Two-dimensional t-SNE projection of mining communication embeddings.

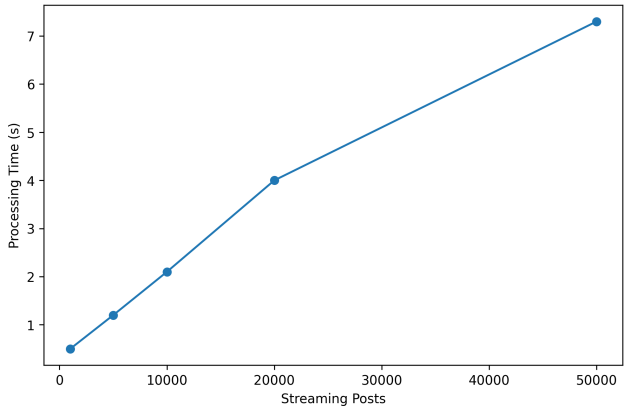


Fig. 12. Runtime scalability under increasing mining communication-stream volume.

C. Mining Semantic Relevance and Drift

Table VII shows substantial variation among mining categories. Safety Monitoring and Underground Operations are the most coherent, with 88% and 86% relevant content, whereas Mineral Market Analytics has only 72% relevance and the highest drift (22%) (Fig. 9 and 10). This result is plausible operationally: safety and underground-operation vocabulary is specialized, while market discussions more readily expand into general economic commentary.

The cosine-similarity distribution in Fig. 3 and the threshold experiment in Fig. 4 jointly explain the relevance decision. The selected 0.70 boundary avoids the lower-threshold regime, which admits weakly related text, and the high-threshold regime, which removes relevant but lexically diverse mining discussions.

D. Cross-Category Robustness and Ablation

Technical categories transfer better than market-oriented categories, reinforcing the role of vocabulary stability. The ablation results are TF–IDF only (accuracy 0.84, F1 0.82), transformer only (0.89, 0.88), and hybrid framework (0.95, 0.93). Importantly, the hybrid row represents the end-to-end acceptance decision, while the RF row in Table VI evaluates the advertisement module alone; the equal rounded accuracy/F1 values therefore describe different tasks and should not be interpreted as evidence that the semantic module is redundant (Table VIII).

The t-SNE visualization is used only as qualitative support: operational, advertisement, off-topic, and drift groups form visibly distinct regions, but quantitative conclusions are based on the held-out metrics rather than on the projection itself (Fig. 11).

E. Scalability and Mining Deployment

Runtime increases approximately linearly over the tested stream sizes in Fig. 12, from about 0.5 s for the smallest batch to about 7.3 s at 50,000 posts. Transformer embedding generation remains the principal computational cost, whereas TF–IDF classification is comparatively lightweight.

The resulting pipeline is suitable for mining operational-

TABLE VII. SEMANTIC RELEVANCE RESULTS ACROSS MINING OPERATIONAL CATEGORIES

Mining category	Relevant (%)	Off-topic (%)	Advertisement (%)	Drift (%)
Safety Monitoring	88	7	5	8
Equipment Maintenance	84	10	6	9
Environmental Monitoring	81	12	7	14
Mineral Market Analytics	72	18	10	22
Operational Logistics	79	13	8	11
Underground Operations	86	9	5	7

TABLE VIII. CROSS-CATEGORY ROBUSTNESS

Transfer pair	Acc.	F1
Safety → Equipment	.89	.88
Equipment → Underground	.91	.90
Environment → Market	.82	.80
Logistics → Safety	.86	.84
Market → Environment	.79	.77

intelligence dashboards, safety monitoring, environmental incident tracking, equipment-maintenance intelligence, and communication-quality monitoring. In this context, "trustworthy" does not imply a universal guarantee; it denotes the measurable reduction of two identified sources of analytical contamination—semantic irrelevance and promotional intrusion—under the reported operating conditions.

F. Comparison with Prior Research and Limitations

Compared with mining-AI studies centered on sensing, mineral processing, predictive maintenance, automation, and communication reliability [4], [11], [13], [14], [15], [18], [21], this study targets unstructured mining communication streams. Compared with general sentence-embedding research [1], [2], it couples semantic alignment with an explicit promotional-content gate. The contribution is therefore an application-oriented hybridization and evaluation framework rather than a new transformer architecture.

The remaining limitations are the dependence on manually defined mining descriptors, evolving industrial vocabulary, computational cost of contextual embeddings, and the need for broader external validation across organizations and regions. These limitations bound the interpretation of the reported results and motivate future deployment studies.

VII. CONCLUSION AND FUTURE WORK

This study presented a mining-specific hybrid NLP framework that combines contextual semantic relevance with supervised advertisement detection to improve the reliability of trend analytics from noisy mining communication streams. On the 12,500-entry corpus, Random Forest and SVM produced the strongest advertisement-detection results (F1 0.93 and 0.92), while the semantic module showed clear category-dependent behavior: Safety Monitoring and Underground Operations exceeded 85% relevance, whereas Mineral Market Analytics exhibited the highest semantic drift at 22%. Threshold analysis selected $\theta = 0.70$ with an F1-score of 0.93, and the end-to-end hybrid ablation reached 0.95 accuracy and 0.93 F1.

The results show that mining communication analytics benefits from separating two different reliability problems: whether a post is semantically aligned with an operational mining topic and whether it contains promotional intent. Future work will focus on externally validating the framework with additional mine-site communication sources, learning topic descriptors automatically, adapting thresholds under vocabulary drift, and evaluating lighter sentence-embedding models for edge and near-real-time deployment.

DATA AVAILABILITY

The de-identified research corpus, annotation guide, mining-topic descriptors, preprocessing configuration, and experiment scripts are available from the corresponding author upon reasonable request, subject to the terms of the source platforms and applicable privacy constraints. Raw platform identifiers and personally identifying metadata are not redistributed.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through project number "NBU-FFMRA-2026-2443-01".

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[3] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of Industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Art. 3159805, 2016, doi: 10.1155/2016/3159805.

[4] J. T. McCoy and L. Auret, "Machine learning applications in minerals processing: A review," *Minerals Engineering*, vol. 132, pp. 95–109, 2019, doi: 10.1016/j.mineng.2018.12.004.

[5] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," *Information*, vol. 13, no. 2, Art. 83, 2022, doi: 10.3390/info13020083.

[6] P. Gundecha and H. Liu, "Mining social media: A brief introduction," in *INFORMS Tutorials in Operations Research*, 2014, pp. 1–17, doi: 10.1287/educ.1120.0105.

[7] O. Zhironkina and S. Zhironkin, "Technological and intellectual transition to Mining 4.0: A review," *Energies*, vol. 16, no. 3, Art. 1427, 2023, doi: 10.3390/en16031427.

[8] E. Lund, A. Pekkari, J. Johansson, et al., "Mining 4.0 and its effects on work environment, competence, organisation and society–a scoping review," *Mineral Economics*, vol. 37, pp. 827–840, 2024, doi: 10.1007/s13563-024-00427-0.

[9] M. A. Akbar, A. A. Khan, S. Mahmood, et al., "Trustworthy artificial intelligence: A decision-making taxonomy of potential challenges," *Software: Practice and Experience*, vol. 54, no. 9, pp. 1621–1650, 2024, doi: 10.1002/spe.3216.

[10] D. Li, S. Liu, B. Wang, C. Yu, P. Zheng, and W. Li, "Trustworthy AI for human-centric smart manufacturing: A survey," *Journal of Manufacturing Systems*, vol. 78, pp. 308–327, 2025, doi: 10.1016/j.jmsy.2024.11.020.

[11] R. Mahmoudi Kouhi, A. T. Najeeb, R. Taherdangkoo, F. D. Ardejani, and C. Butscher, "A survey study on the adoption and perception of artificial intelligence in the mining industry," *Discover Applied Sciences*, vol. 7, Art. 713, 2025, doi: 10.1007/s42452-025-07342-1.

[12] R. Osei, S. Frimpong, and A. Venkat, "Human-machine collaboration in mining: A critical review of emerging frontiers of intelligence systems in the mining industry," *The Extractive Industries and Society*, vol. 24, Art. 101746, 2025, doi: 10.1016/j.exis.2025.101746.

[13] H. Yang, W. Feng, H. Diao, and S. Xia, "Artificial intelligence transforming minerals engineering: Key trends in literature and applications," *Minerals Engineering*, vol. 234, Art. 109741, 2025, doi: 10.1016/j.mineng.2025.109741.

[14] L. Reyes-Bozo, E. Vyhmeister, H. Valdés-González, G. G. Castane, J. C. Vidal, J. E. Martínez-Hernández, and E. Villarroel-Utreras, "Artificial intelligence in the copper mining industry: A systematic mapping review and qualitative synthesis," *Processes*, vol. 14, no. 16, Art. 2540, 2026, doi: 10.3390/pr14162540.

[15] J. L. Arenas Ñiquin et al., "Machine learning in smart mining: A systematic review of applications, algorithms, benefits, and challenges," *Algorithms*, vol. 19, no. 3, Art. 197, 2026, doi: 10.3390/a19030197.

[16] Z. Guo, G. Liu, S. Chai, W. Liu, and F. Jiang, "Implementation and intelligent path exploration of multimodal large model architecture for open-pit coal mine production," *Results in Engineering*, vol. 29, Art. 109283, 2026, doi: 10.1016/j.rineng.2026.109283.

[17] C. C. Corrigan and S. A. Ikonnikova, "A review of the use of AI in the mining industry: Insights and ethical considerations for multi-objective optimization," *The Extractive Industries and Society*, vol. 17, Art. 101440, 2024, doi: 10.1016/j.exis.2024.101440.

[18] A. Gharbi, M. Ayari, Y. El Touati, Z. Klai, M. S. Elsayed, and E. M. Abdelrahim, "Prediction of mining-induced subsidence in Saudi Arabia phosphate mines using ANN method," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 8, 2025, doi: 10.14569/IJACSA.2025.0160852.

[19] A. Gharbi, M. Ayari, N. Albalawi, Y. El Touati, and Z. Klai, "Intelligent HVAC control: Comparative simulation of reinforcement learning and PID strategies for energy efficiency and comfort optimization," *Mathematics*, vol. 13, no. 14, Art. 2311, 2025, doi: 10.3390/math13142311.

[20] A. Gharbi, M. Ayari, N. Ben Halima, A. Elkamel, and Z. Klai, "Fairness criteria in multi-agent systems: Optimizing autonomous traffic management through the hierarchical Stackelberg strategy," *Applied Sciences*, vol. 15, no. 13, Art. 6997, 2025, doi: 10.3390/app15136997.

[21] M. Ayari, A. Gharbi, Z. Klai, A. Hasnaoui, M. S. Elsayed, and E. Abdelrahim, "Game-Theoretic Electromagnetic Isolation Framework for Wireless Connectivity in Mineral and Phosphate Mining Environments," *European Journal of Pure and Applied Mathematics*, vol. 18, no. 4, Art. 7083, 2025, doi: 10.29020/nybg.ejpam.v18i4.7083.

MedEdge-Conf: A Lightweight Confidence-Scoring Framework for Edge-Deployed LLMs in IoMT Health Monitoring

Fahd Koraiche[1]*, Rachid Dehbi[2], Amine Dehbi[3]
Research Laboratory of Data Engineering and Intelligent Systems Laboratory (LIDSI),
Faculty of Science Aïn Chock, Casablanca, Morocco[1,2]
Research Laboratory of Mathematics and Computer Science (LTI), Faculty of Sciences Ben M'Sik, Casablanca, Morocco[3]

***Abstract*—Remote patient monitoring systems based on the Internet of Medical Things (IoMT) increasingly use large language models (LLMs) to convert vital signs into readable alerts, but lightweight edge-deployed LLMs may hallucinate, contradict abnormal vitals, or overstate confidence. This study proposes MedEdge-Conf, a lightweight confidence-scoring framework combining physiological coherence aligned with the National Early Warning Score 2 (NEWS2), token-level entropy, and semantic stability into a calibrated composite score driving a deliver/review/escalate decision. We evaluate MedEdge-Conf on 249 test scenarios from two open-access PhysioNet resources (CinC2019, MIT-BIH), using three edge-deployed models (Phi-2, TinyLlama, Gemma-2B, all `Q5_K_M`), with ground truth from two adjudicated human raters per model. Calibrated fusion reduces the false acceptance rate (FAR) of dangerous hallucinations from 100 percent under naive delivery to 26.4 percent for Phi-2, 17.3 percent for TinyLlama, and 32.1 percent for Gemma-2B, with AUROC improvement over naive scoring statistically significant in every case (DeLong's test, p less than 0.03); precision on delivered responses remains low (14.0-22.8 percent), so it filters dangerous delivery rather than guaranteeing reliability. An ablation study shows the signal driving this improvement is model-dependent: physiological coherence alone is strongest for Phi-2, token entropy helps TinyLlama, and semantic stability matters most for Gemma-2B. Against three literature baselines, MedEdge-Conf achieves the highest AUROC and lowest FAR for every model. Real Raspberry Pi 5 measurements across nine model-quantization configurations show TinyLlama is fastest and lightest, while Gemma-2B is markedly slower and heavier despite being the safest raw generator.**

***Keywords*—Edge AI; internet of medical things; medical LLM; hallucination detection; confidence scoring**

I. Introduction

Remote patient monitoring has become a central component of digital health systems, particularly for chronic disease follow-up and early detection of physiological deterioration. IoMT devices and wearables continuously collect heart rate, oxygen saturation, respiratory rate, temperature, ECG signals, and activity indicators, but these are useful only once translated into timely, clinically safe information. Recent progress in LLMs enables converting such measurements into natural-language explanations, but a standard cloud-centric architecture creates three limitations in health contexts: latency from transmitting data to remote infrastructure before producing a response, privacy and governance risk from continuous transfer of raw physiological streams to external services, and dependence on connectivity that may be unreliable in home or low-resource settings.

A natural response is to move inference closer to the patient using lightweight quantized LLMs such as Phi-2, TinyLlama, and Gemma variants, executed locally via runtimes such as llama.cpp; quantization can reduce model size and energy requirements, making on-device LLM inference increasingly feasible for embedded applications [6]. Feasibility is not equivalent to safety, however: an edge LLM that labels a critical state as "normal" or "stable" may create direct patient risk, and compression and limited model capacity can worsen instability and hallucination when interpreting abnormal physiological values.

The research gap addressed here is therefore not how to run an LLM on a Raspberry Pi, but how an IoMT system can decide whether a local LLM response is trustworthy enough to show the patient. Existing hallucination-detection methods often rely on large verifier models, retrieval pipelines, or multi-agent debate, not always compatible with an edge gateway's resource budget; generic hallucination detection also does not necessarily capture physiological contradictions such as low SpO2 described as normal, and even token-entropy-based uncertainty estimates can leave a model confidently wrong about a clinical fact [9] (Section II).

This study introduces MedEdge-Conf, a lightweight confidence-scoring framework for edge-deployed LLMs in IoMT health monitoring, scoring each generated alert on three signals (C_1 physiological coherence/NEWS2, C_2 token entropy, C_3 semantic stability) fused into a calibrated composite score that determines whether the alert is sent, reviewed, or rejected/escalated, so no response reaches the patient without reliability assessment.

The contributions of this work are fourfold:

- We propose MedEdge-Conf, among the first confidence-scoring frameworks for medical LLM outputs under edge-deployment constraints, integrating clinical validation, internal uncertainty, and generation stability within a single-board computer's memory and latency budget.

*Corresponding author.

- We align physiological coherence with the NEWS2 early warning system, so the safety layer respects established clinical thresholds rather than ad-hoc heuristics.
- We execute the complete benchmarking pipeline on a physical Raspberry Pi 5, reporting latency, throughput, memory, CPU, and SoC temperature across all nine model-quantization configurations (Section IV-E).
- We conduct ablation studies on human-annotated data from PhysioNet/CinC2019 and MIT-BIH, quantifying each signal's marginal contribution across all three models (Section V-A), and report a baseline comparison against three literature methods (Section IV-D).

The remainder of this study is organized as follows: Section II reviews related work; Section III details the MedEdge-Conf architecture, data sources, scoring components (C_1–C_3), and statistical validation protocol; Section IV reports safety-filtering, ablation, baseline-comparison, and edge-performance results; Section V discusses results and limitations; Section VI concludes.

II. Related Work

A. Physiological Monitoring Datasets and Clinical Scoring

PhysioNet provides open access to physiological databases and software resources for biomedical signal research [1]. MIMIC-III contains bedside physiological signals and vital-sign numerics from intensive care settings [3], with its successor MIMIC-IV extending to broader electronic health record data [15]; the PhysioNet/Computing in Cardiology 2019 Challenge targets early sepsis prediction [4]; and MIT-BIH is a standard annotated ECG benchmark [5].

NEWS2, developed by the Royal College of Physicians, aggregates respiration rate, oxygen saturation, systolic blood pressure, pulse, consciousness, temperature, and supplemental oxygen into a deterioration risk score [2], assigning 0–3 points per parameter and mapping the aggregate to low (0–4), medium (3–6 in any single parameter or aggregate 5–6), or high ($\geq$ 7) risk. MedEdge-Conf uses NEWS2 as a constraint layer, penalizing responses that contradict the NEWS2-derived risk category, rather than replacing NEWS2 or applying arbitrary rules.

B. Edge LLM Deployment and Uncertainty Estimation in Medical LLMs

LLMPi investigates quantization-based optimization for Raspberry Pi [6]; llama.cpp supports local inference across hardware using GGUF quantized models [7]; and Nguyen et al. [11] benchmark LLM inference on single-board computers, informing our experimental protocol. These developments, like broader edge-AI/IoT monitoring architectures generally, establish technical feasibility — local inference, resource constraints, communication overhead — but not output trustworthiness in medical contexts, where uncertainty estimation remains difficult [8] and token-level probabilities estimate confidence without guaranteeing correctness, since a model may be confidently wrong [9]; MedEdge-Conf instead adds a reliability-aware decision layer that checks each response before delivery, rather than optimizing generation or transport alone. Wang et al. [10] propose word-sequence entropy for medical QA, showing correlation between entropy and factual errors.

SelfCheckGPT compares stochastically sampled generations via n-gram overlap, NLI scoring, or LLM prompting to flag hallucinations without model-internal access [17], but is costly and fragile to paraphrase. Broader LLM uncertainty-quantification surveys categorize methods by cost and by reliance on internal signals versus external, consistency-based verification [22]. Guardrail toolkits such as NeMo Guardrails [23] enforce runtime constraints between an application and a general-purpose LLM; MedEdge-Conf is a domain-specific, resource-constrained instance of this paradigm, specialized for physiological alerting rather than open-domain dialogue safety.

Lee et al. [12] highlight the gap between internal model confidence and clinical accuracy, and Rajpurkar et al. [13] emphasize the need for rigorous safety and reliability evaluation of AI in medicine. MedEdge-Conf addresses these gaps with a lightweight hybrid score combining uncertainty signals and physiological coherence for near-real-time, local execution on resource-constrained edge hardware, rather than cloud-based question answering or cloud-scale ensembles.

C. Clinical Safety Assessment, Linguistic Robustness, and Human Factors in Alert Systems

Asgari et al. [24] measure hallucination rates and clinical safety in LLM-generated medical summaries via structured clinician review categorizing errors by severity — an approach MedEdge-Conf shares through its four-way reliability taxonomy (Section III-B) — but target retrospective, cloud-hosted summarization rather than the latency and compute budget of an edge gateway.

NegEx, a curated list of negation trigger phrases determining whether a clinical finding is asserted or negated, remains widely used for its simplicity despite known failure modes on complex or multiply negated sentences [18]. C_1's keyword-based contradiction detection (Section III-E) is conceptually related but has not been evaluated for robustness to negation or hedged language, a limitation revisited in Section V-C.

MedEdge-Conf's review tier routes uncertain cases to a human and so inherits the well-documented risk of alert fatigue from interruptive clinical alerts [25], discussed further in Section V-C.

III. Methodology

A. System Architecture

The proposed system targets an Edge-IoMT architecture for home or bedside monitoring: physiological measurements from sensors or wearables are transmitted to a local gateway (a Raspberry Pi 5, 8 GB RAM), which preprocesses, constructs a compact prompt, runs a quantized local LLM, evaluates the response via MedEdge-Conf, and routes the output by confidence level, with an optional cloud/hospital escalation layer. The pipeline comprises six stages: 1) sensor data acquisition and local validation; 2) prompt construction with structured vital signs; 3) LLM response generation with log-probability extraction; 4) confidence scoring via MedEdge-Conf; 5) decision routing based on calibrated thresholds; and

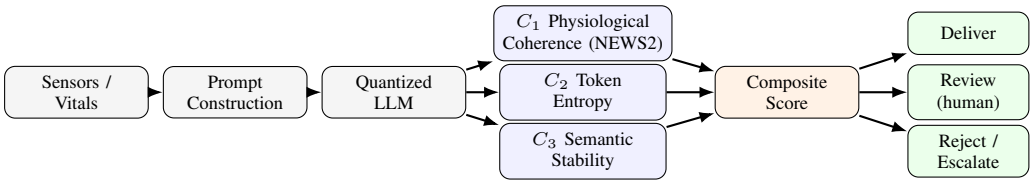


Fig. 1. MedEdge-Conf pipeline: Local LLM output is scored by C_1/C_2/C_3 (Sections III-E–III-G) and fused into a calibrated composite score (Section III-H) driving the delivery decision.

6) optional cloud escalation for rejected responses. Fig. 1 summarizes stages 3–6.

B. Data Sources and Scenario Construction

The experimental protocol uses two Open Access PhysioNet resources: the PhysioNet/Computing in Cardiology 2019 sepsis dataset (CC-BY 4.0, no credentialing) [4], providing real per-hour vital-sign records (heart rate, SpO2, respiratory rate, systolic blood pressure, temperature, and fraction of inspired oxygen) for tens of thousands of ICU patients, and the MIT-BIH Arrhythmia Database (Open Data Commons Attribution License, no credentialing) [5], providing real annotated ECG recordings with beat-level rhythm labels. MIMIC-III's waveform database [27] (distinct from the MIMIC-III clinical database [3] cited in Related Work as background) was not extracted, since it requires PhysioNet credentialing not pursued in this study.

We constructed 1,247 structured patient scenarios with vital signs and, where available, a trend descriptor, sourced per severity category in this preference order: 1) real CinC2019 patient-hours — heart rate, SpO2, respiratory rate, systolic BP, and temperature are actual recorded values, supplemental-oxygen is derived from FiO2, and trend descriptors come from within-patient sequential comparison against preceding hours, not templated text; 2) real MIT-BIH-derived scenarios — heart rate is computed from consecutive R-peak intervals ($60/$RR-interval-seconds at 360 Hz), the trend descriptor comes from the WFDB beat-annotation symbol, and the remaining ECG-only gaps (SpO2, respiratory rate, systolic BP, temperature, consciousness) are filled from a documented synthetic range profile matched to beat severity; 3) fully-synthetic backfill from the same range profiles if the two real sources cannot supply enough eligible rows. No fully-synthetic backfill was needed: every scenario's primary vital(s) — CinC2019's full panel, or MIT-BIH's ECG-derived heart rate — come from a real recorded source, though MIT-BIH scenarios' non-ECG vitals remain profile-filled as noted above — stable: 423 (360 CinC2019, 63 MIT-BIH); borderline: 412 (401 CinC2019, 11 MIT-BIH); critical: 412 (350 CinC2019, 62 MIT-BIH) — with a minimum MIT-BIH share reserved per category. Consciousness (AVPU), unrecorded in either real source, defaults to "Alert." Forcing every scenario's AVPU to non-Alert instead shifts the NEWS2 risk category for 152/249 (61.0%) test scenarios (7 low→high, 77 low→medium, 68 medium→high) — not a minor simplification, though the true consciousness state is unrecorded in both sources and its concordance with "Alert" cannot be established from this data.

Each response was annotated into one of four categories, independently, by two raters (the first author and a second annotator):

- Reliable: clinically appropriate and consistent with vital signs.
- Incomplete: partially correct but missing critical information.
- Contradictory: contains statements that conflict with physiological evidence.
- Unsafe: provides false reassurance in the presence of abnormal vital signs.

Inter-rater agreement, computed with Cohen's kappa [19] on the two raters' independent labels before reconciliation, was $\kappa = 0.594$ (validation, $n = 249$) and $\kappa = 0.585$ (test, $n = 249$), both "moderate agreement" on the Landis and Koch scale; 69/249 validation items and 70/249 test items (27.7% and 28.1%) received different labels. Rather than discard these or default to one rater, each was resolved by joint review of the vitals, NEWS2 category, and response text to a single final label — the adjudicated label, not either rater's original, is the ground truth used throughout Section IV. This two-stage protocol is standard for subjective clinical-text annotation and lets κ measure the task's subjectivity directly (a limitation discussed in Section V-C).

Each scenario is also assigned one of three physiological categories — stable, borderline, or critical — a controlled grouping variable, not the reliability ground truth (which comes from the four-way annotation above). The dataset was split into training (60%, $n = 749$), validation (20%, $n = 249$), and test (20%, $n = 249$) sets, stratified by physiological category and data source.

C. Local LLM Configuration

The edge LLM layer uses lightweight quantized GGUF models — Microsoft Phi-2 (2.7B), TinyLlama-1.1B, and Google Gemma-2B — run through llama-cpp-python with a 512-token context window reflecting Raspberry Pi-class memory and latency constraints, decoding temperature 0.3 for stable responses, and output limited to 128 tokens for short alert generation.

We evaluate three quantization levels: `Q4_0` (4-bit), `Q5_K_M` (5-bit), and `Q8_0` (8-bit) — `Q4_0` prioritizes speed and memory efficiency, `Q8_0` accuracy, and `Q5_K_M` a middle ground. For each model-quantization pair, we measure generation latency, tokens per second, RAM consumption, and CPU utilization.

A typical prompt contains the patient's vital signs and asks the model to produce a brief assessment and alert level, e.g., *"Patient vital signs: Heart rate 118 bpm, SpO2 87%,*

Temperature 39.2°C, Respiratory rate 28/min, Systolic BP 95 mmHg. Generate a short non-diagnostic alert using categories: normal, warning, or critical." The generated response is never delivered directly; it is first evaluated by MedEdge-Conf.

D. MedEdge-Conf Composite Score

MedEdge-Conf computes a composite score from three components:

$$S = \alpha C_1 + \beta C_2 + \gamma C_3 \tag{1}$$

where, C_1 (physiological coherence: contradiction between text and NEWS2-aligned vitals; very low edge cost), C_2 (token entropy: uncertainty from output log-probabilities; low edge cost), and C_3 (semantic stability: sentence-embedding cosine similarity; moderate edge cost) are normalized to $[0, 1]$. In the initial heuristic configuration, $\alpha = 0.50$, $\beta = 0.25$, and $\gamma = 0.25$, fixed a priori per the design rationale detailed in the following subsections. These weights are subsequently calibrated using logistic regression on the validation set (yielding w_1, w_2, w_3, Section III-H), with the three normalized components as input features to a calibrated probability of reliability, the final composite score driving the deliver/review/escalate decision.

E. C_1: Physiological Coherence Score

C_1 verifies response consistency with the raw vital signs through a two-stage process.

Stage 1: NEWS2 Score Calculation. For each scenario, the NEWS2 aggregate score is computed from the vital signs (respiration rate, SpO2, systolic blood pressure, pulse, consciousness, temperature, supplemental oxygen) and mapped to the risk category defined in Section II-A.

Stage 2: Linguistic Contradiction Detection. We parse the LLM response for keywords indicating reassurance ("normal", "stable", "within range", "satisfactory", "no concern", "fine") or alarm ("critical", "warning", "abnormal", "concerning", "seek help", "urgent"). If the NEWS2 category is medium or high but the response contains reassuring terms, C_1 applies a strong penalty; if NEWS2 is low but the response contains alarm terms without justification, a milder penalty avoids unnecessary anxiety. C_1 is bounded between 0 and 1, where 1 means no contradiction and 0 means a severe contradiction.

This component is deliberately interpretable — when C_1 lowers the score, the system reports the specific reason, e.g., "NEWS2=7 (high risk) but response states 'stable' " or "SpO2 84% described as normal" — making it easier to audit than a purely neural verifier, and computationally cheap, requiring only dictionary lookups and arithmetic suitable for Raspberry Pi 5.

Stage 2a's keyword classifier has known limitations: negated or hedged constructions (e.g., "no longer stable") are not reliably handled and can invert the contradiction signal if left unaddressed [18]. Testing on annotated data exposed a second gap: Stage 2a compares keywords only to the *overall* risk category and misses checkable single-parameter claims, such as "systolic BP is above normal (126 mmHg)" when 126 mmHg is within the NEWS2 normal band (111–219 mmHg) — an error that need not change the overall category or involve a keyword mismatch.

We therefore added Stage 2b: a per-parameter numeric contradiction check that scans the response (via parameter-name and directional-claim regular expressions, e.g. "heart rate ... above normal") for an explicit directional claim (high/low/normal) on each of the five core vitals, compares it against that parameter's true direction computed from the value and the NEWS2 Scale 1 normal band, and applies Stage 2a's severe penalty on any mismatch (normal claimed abnormal, abnormal claimed normal, or wrong direction), independent of the overall NEWS2 category. Like Stage 2a, this regex/keyword-window approximation can miss indirect phrasing or unusual sentence structure.

On the annotated test set ($n = 249$), the combined C_1 detector (Stages 2a+2b) achieves standalone precision=74.4% and recall=79.2% against the ground-truth "unsafe or contradictory" label (adjudicated two-annotator ground truth, Section III-B), independent of the downstream composite score. A negation-aware layer (e.g., NegEx [18] or a lightweight learned classifier), and extending Stage 2b for co-occurring reassurance/alarm keywords that currently cancel out under Stage 2a's logic, are future work (Section V-C).

F. C_2: Token Entropy Score

C_2 estimates the uncertainty of the LLM response from token-level log-probabilities. For each generated token, the probability distribution over the vocabulary is converted into Shannon entropy:

$$H_t = -\sum_i p_i \log(p_i) \tag{2}$$

where, p_i is the probability of the i-th candidate token at position t. The mean entropy over the response is normalized into a confidence score using a sigmoid transform calibrated on the validation set:

$$C_2 = \frac{1}{1 + \exp\left(k\left(H_{mean} - H_0\right)\right)} \tag{3}$$

where, H_0 is the median validation-set entropy and $k = 2.0$ is a steepness parameter. Higher entropy indicates less certainty about the next-token decision. C_2 reuses information already produced during inference, needing no additional model, but a response can be low-entropy yet medically wrong, so it is combined with C_1 and C_3 rather than used standalone.

G. C_3: Semantic Stability Score

C_3 estimates response stability by comparing the delivered response (temperature=0.3) against one additional generation for the same prompt at temperature=0.7 (perturbed), using semantic similarity from lightweight sentence embeddings from the all-MiniLM-L6-v2 model (22M parameters), which runs efficiently on Raspberry Pi 5 via ONNX Runtime.

Semantic similarity is the cosine similarity between the two embedding vectors:

$$C_3 = \frac{e_1 \cdot e_2}{\|e_1\| \, \|e_2\|} \tag{4}$$

High similarity suggests stable generation; low similarity suggests sensitivity to decoding variation. Unlike Jaccard similarity over tokens, cosine similarity over embeddings captures semantic equivalence across different wording [16].

Where, the two responses differ in length by more than 50%, a length-normalization factor prevents trivial mismatches from dominating the score. The output is bounded to $[0, 1]$.

H. Score Calibration and Decision Policy

The three components are normalized, then combined via two fusion strategies: 1) heuristic fusion, fixed weights $\alpha = 0.50$, $\beta = 0.25$, $\gamma = 0.25$; and 2) calibrated fusion, logistic regression trained on the validation set to predict "reliable vs. unreliable" (unreliable includes contradictory and unsafe responses).

The logistic regression model learns optimal weights and a bias term:

$$P(\text{reliable}) = \sigma\left(w_1 C_1 + w_2 C_2 + w_3 C_3 + b\right) \tag{5}$$

On the validation set ($n = 249$, adjudicated ground truth, Section III-B), the calibrated weights converged to $w_1 = 2.393$, $w_2 = 0.319$, $w_3 = -0.129$, bias $= -1.549$, confirming the dominance of physiological coherence but revealing an unexpected pattern: w_3 (semantic stability, C_3) is negative — higher stability between the two temperature samples was associated with *lower* reliability. A plausible hypothesis (Section V-A) is that a confidently wrong response (e.g., "vital signs are normal") tends to repeat nearly verbatim across both samples, while a more nuanced response shows natural wording variation. 5-fold cross-validation confirms each weight's sign is stable across folds ($w_1 = 2.345 \pm 0.161$, $w_2 = 0.292 \pm 0.121$, $w_3 = -0.120 \pm 0.191$, bias $= -1.510 \pm 0.180$), though w_3's magnitude is small relative to its fold-to-fold standard deviation.

The calibrated confidence score controls a three-tier routing policy: deliver (score ≥ 0.43), review ($0.40 \leq$ score < 0.43), and reject/escalate (score < 0.40). These thresholds were selected by grid search over deliver/reject candidates in $[0.05, 0.95]$ at 0.01 resolution (the same grid used for every threshold-fitting step in this study) to maximize F2-score on the validation set, prioritizing recall of unsafe/contradictory responses over precision, subject to a minimum review-band width of 0.02 enforced for all three models so this band cannot collapse to near-zero width regardless of the unconstrained F2 optimum (motivated by the narrow-review-band limitation discussed in Section V-C); on the test set, the review band now contains 1 of 249 scenarios (Section IV-F, Case 1). A parallel threshold pair was fit for the heuristic-fusion score by the same procedure (deliver ≥ 0.54, reject < 0.52), since the two scores are on different scales.

Beyond discrimination metrics (AUROC, AUPRC), we assess whether $P(\text{reliable})$ is well-calibrated statistically — i.e., whether a predicted probability of roughly 0.7 corresponds empirically to reliability about 70% of the time — using Expected Calibration Error and Brier score [20], reported in Section IV-B.

I. Statistical Validation Protocol

All point estimates in Section IV (FAR, AUROC, AUPRC, precision, recall, ECE, Brier score) are computed on the $n = 249$ test set (Section III-B), with 95% confidence intervals via nonparametric bootstrap resampling (2,000 resamples) and DeLong's test [21] for AUROC-difference significance between scoring configurations. Bootstrap CIs and DeLong's test cover all pairwise comparisons discussed in the text: Table I (Section IV-B), Table II (Section IV-C), and Table III (Section IV-D). Because DeLong's test compares AUROC rather than the thresholded delivery decision FAR is computed from, paired McNemar's exact tests supplement it for the FAR comparisons in Section IV-B. Since each table reports many pairwise tests, we applied Benjamini–Hochberg FDR correction ($q = 0.05$) within each table's DeLong-test family (Tables I–III) and within Table I's heuristic-vs-calibrated McNemar tests; every significance conclusion in Section IV is unchanged after correction. 5-fold cross-validation of the calibrated weights (w_1, w_2, w_3) was completed for all three models (Sections III-H, IV-G, IV-H).

IV. Experimental Results

A. Dataset and Annotation Statistics

The test set comprises 249 scenarios: 84 stable, 82 borderline, and 83 critical, identical across all three evaluated models (Phi-2, TinyLlama, and Gemma-2B, all `Q5_K_M`). Sections IV-B–IV-D report results for all three models together; Sections IV-G–IV-H give each replicated model's stats and weights. The Phi-2 LLM (a base model — Microsoft has not released an instruction-tuned Phi-2, discussed as a limitation in Section V-C) generated responses distributed, under the final adjudicated ground truth (Section III-B), as follows: 21 reliable, 103 incomplete, 84 contradictory, and 41 unsafe. Only 8.4% of responses were judged reliable — low enough that a safety layer is a precondition, not optional, for any patient-facing use.

B. Safety Filtering Performance

Table I presents the primary safety metrics for all three evaluated models, comparing MedEdge-Conf's heuristic and calibrated fusion on the human-annotated test set.

The false acceptance rate (FAR) measures the proportion of contradictory or unsafe responses incorrectly classified as reliable and delivered. Calibrated fusion reduces FAR to 26.4% for Phi-2 (73.6% relative reduction), 17.3% for TinyLlama (82.7%), and 32.1% for Gemma-2B (67.9%), with AUROC improvement over naive scoring statistically significant for every model (DeLong's test: Phi-2 $z = 2.219$, $p = 0.027$; TinyLlama $z = 3.839$, $p = 0.0001$; Gemma-2B $z = 4.299$, $p = 1.7 \times 10^{-5}$); heuristic fusion's AUROC gain over naive is also significant for every model ($p \leq 0.016$). Precision is low in absolute terms across all models (10–23%) because "reliable" is a rare outcome (8.4%/8.0%/14.9% prevalence for Phi-2/TinyLlama/Gemma-2B): even a well-calibrated filter cannot reach high precision against a class this rare without

TABLE I. SAFETY FILTERING PERFORMANCE ON THE TEST SET, ALL THREE MODELS ($n = 249$)

	Heuristic			Calibrated		
Metric	**Phi-2**	**TinyLlama**	**Gemma-2B**	**Phi-2**	**TinyLlama**	**Gemma-2B**
FAR (%, 95% CI)	24.8 (17.2–32.5)	60.4 (52.2–68.6)	32.1 (23.4–41.2)	26.4 (18.9–34.2)	17.3 (11.1–23.9)	32.1 (23.7–41.3)
Unsafe delivered	11	11	0	11	2	1
Contradictory delivered	20	73	34	22	22	33
Precision (reliable)	13.3%	10.6%	21.8%	14.8%	14.0%	22.8%
Recall (reliable)	81.0%	100.0%	73.0%	95.2%	75.0%	83.8%
F1-score	22.8%	19.1%	33.5%	25.6%	23.6%	35.8%
AUROC (95% CI)	0.610 (0.512–0.698)	0.729 (0.613–0.831)	0.666 (0.578–0.753)	0.596 (0.509–0.676)	0.664 (0.577–0.747)	0.674 (0.593–0.753)

Naive delivery (all responses delivered) has FAR=100.0% (100.0–100.0) and AUROC=0.500 (0.500–0.500) for every model by construction; naive precision equals each model's reliable-response prevalence (Phi-2 8.4%, TinyLlama 8.0%, Gemma-2B 14.9%, Sections IV-A, IV-G, IV-H), recall is 100.0%, F1 is 15.6%/14.9%/25.9%. AUPRC (heuristic/calibrated): 0.104/0.098 (Phi-2), 0.198/0.112 (TinyLlama), 0.229/0.223 (Gemma-2B); naive AUPRC equals each model's reliable-response prevalence. ECE (heuristic/calibrated): 0.411/0.398, 0.604/0.383, 0.322/0.390; naive ECE is 0.916/0.920/0.851. Brier score (heuristic/calibrated): 0.287/0.270, 0.455/0.239, 0.260/0.289; naive Brier is 0.916/0.920/0.851. Calibrated-fusion review-tier counts (of 249, min. 0.02-wide band, Section III-H): Phi-2 1; TinyLlama 25; Gemma-2B 8.

being extremely conservative, at the cost of recall. (A paired McNemar's exact test likewise confirms naive vs. calibrated differs significantly for every model, $p < 10^{-20}$, trivially so since naive delivers every dangerous response by construction.)

Whether calibration improves on the fixed heuristic weights, however, is model-dependent. For Phi-2, it does *not* clearly help: heuristic fusion has a numerically lower FAR (24.8% vs. 26.4%) and higher AUROC (0.610 vs. 0.596), though the difference is not significant by AUROC (DeLong's test, $z = 0.357$, $p = 0.721$) or by McNemar's exact test on the delivery decision ($p = 0.50$). For TinyLlama, calibration is a large, unambiguous improvement, cutting FAR from 60.4% to 17.3%, confirmed by McNemar's exact test ($p = 2.7 \times 10^{-17}$). For Gemma-2B, heuristic and calibrated fusion land on an *identical* FAR (32.1%) because their deliver thresholds happen to bind on the same decision boundary (McNemar's exact test, $p = 1.00$); calibration's only measurable advantage there is in recall (83.8% vs. 73.0%) and F1 (35.8% vs. 33.5%). The practical value of data-driven calibration therefore varies by base model, discussed further in Section V-A.

Because the test set is CinC2019-majority (222/249 vs. 27/249 MIT-BIH), we checked whether calibrated-fusion performance differs by source. It does for Phi-2: AUROC drops from 0.616 to 0.340 (below chance) on MIT-BIH, significant by bootstrap on independent samples ($p = 0.013$), plausibly because MIT-BIH's non-ECG vitals are synthetic profile-fills C_1 may not handle as reliably as CinC2019's real panel. TinyLlama (0.682 vs. 0.600) and Gemma-2B (0.687 vs. 0.625) show the same direction but not significance at $n = 27$ ($p = 0.514$, $p = 0.800$).

C. Ablation Study

Table II presents the ablation study across all three models, comparing single-component and partial-combination configurations against the full MedEdge-Conf system on the same test set as Table I.

The three models show three distinct patterns, reported directly rather than reconciled into one story. For Phi-2, C_1 (physiological coherence) alone is the strongest configuration (AUROC=0.663) and *significantly outperforms* the full calibrated fusion (AUROC=0.596; DeLong's test, $z = 2.841$, $p = 0.0045$), though not heuristic fusion ($z = 1.627$, $p = 0.104$); $C_1 + C_3$ has the single highest point estimate of any configuration for any model (AUROC=0.675), also significantly ahead of the calibrated fusion ($z = 3.008$, $p = 0.0026$) but not of C_1 alone ($z = 0.468$, $p = 0.640$). C_2 (token entropy) alone scores *below* chance for Phi-2 (AUROC=0.315). For TinyLlama, the opposite holds: the full heuristic fusion (AUROC=0.729) *significantly outperforms* C_1 alone (AUROC=0.607; $z = 2.253$, $p = 0.024$), and C_2 is a genuinely useful signal (AUROC=0.625) rather than a harmful one; the calibrated fusion (AUROC=0.664) is not significantly different from C_1 alone ($z = 1.391$, $p = 0.164$) or heuristic fusion ($z = 0.925$, $p = 0.355$). For Gemma-2B, a third pattern emerges: the full calibrated fusion (AUROC=0.674) again *significantly outperforms* C_1 alone (AUROC=0.635; $z = 2.716$, $p = 0.0066$), but here the improvement is driven by C_3, not C_2 ($C_1 + C_3$ AUROC=0.675 is not significantly different from the full fusion, $z = 0.074$, $p = 0.941$, but is significantly better than C_1 alone, $z = 2.349$, $p = 0.019$); C_2 alone (AUROC=0.588) is neither clearly harmful nor clearly useful. C_3 is thus weak-to-harmful for Phi-2 and TinyLlama (AUROC 0.553 and 0.367) but is Gemma-2B's single strongest individual component (AUROC=0.639) — a third distinct role for the same signal across three models, interpreted further in Section V-A.

TABLE II. ABLATION STUDY AUROC, ALL THREE MODELS (TEST SET, $n = 249$)

Configuration	**Phi-2**	**TinyLlama**	**Gemma-2B**
C_1 only	0.663	0.607	0.635
C_2 only	0.315	0.625	0.588
C_3 only	0.553	0.367	0.639
$C_1 + C_2$	0.580	0.681	0.651
$C_1 + C_3$	0.675	0.505	0.675
$C_2 + C_3$	0.323	0.593	0.676
Heuristic (all 3)	0.610	0.729	0.666
Calibrated (all 3)	0.596	0.664	0.674

FAR values for the partial-component rows vary by model and configuration (Section III-H) rather than the discrimination shown here; only the full heuristic/calibrated rows are used for delivery decisions, and their FAR/precision/recall match Table I.

C_1's standalone precision/recall against the adjudicated ground truth also varies sharply by model: 74.4%/79.2% for Phi-2, 98.0%/34.5% for TinyLlama (near-perfect precision but low recall, since TinyLlama's contradictions are disproportionately driven by fabricated non-vital details — invented medical history, medication, or diagnoses — a failure mode C_1's

keyword/numeric-contradiction design was not built to catch), and 48.7%/72.6% for Gemma-2B (a more heterogeneous mix of the numeric-mismatch pattern C_1 targets and other failure types). Rows without C_1 are near or below chance for Phi-2 and TinyLlama but not Gemma-2B, showing physiological grounding is not equally load-bearing across models.

D. Baseline Comparison

We compare MedEdge-Conf against three baseline approaches from the literature: 1) Token-Entropy Only, a threshold on mean token entropy [9]; 2) SelfCheckGPT, sampling-based consistency with Jaccard similarity [17]; and 3) Semantic Entropy, entropy over semantic clusters of multiple generations [26], approximated here via embedding-based clustering rather than the original NLI-based bidirectional entailment clustering, since implementing a full NLI model was out of scope for an edge-resource-constrained comparison (documented in the implementation and revisited in Section V-C).

MedEdge-Conf has the highest point-estimate AUROC and lowest FAR of all four methods for every model, but the strength of its advantage is model-dependent. For Phi-2, MedEdge-Conf significantly outperforms only Token-Entropy Only ($p = 0.0003$) — the same signal Section IV-C already showed performs below chance in isolation (C_2 only, AUROC=0.315, since both reuse the identical entropy computation); its margin over the two sampling-based baselines does not reach significance at $n = 249$ ($p = 0.208$ and $p = 0.202$). For TinyLlama and Gemma-2B, MedEdge-Conf significantly outperforms all three baselines, including both sampling-based ones (TinyLlama: $p = 0.0062$ and $p = 0.0004$; Gemma-2B: $p = 0.00037$ and $p = 1.6 \times 10^{-6}$). Token-Entropy Only itself performs below chance for Phi-2 (AUROC=0.315) and Gemma-2B (AUROC=0.412) but is a reasonably strong baseline for TinyLlama (AUROC=0.625), consistent with Table II's finding that C_2's usefulness varies by model. SelfCheckGPT and Semantic Entropy perform at or below chance for TinyLlama (0.427, 0.410) and Gemma-2B (0.481, 0.421) — the weakest baseline showing of the three models — strengthening the case for grounding the confidence score in NEWS2-aligned physiological rules over generic language-model uncertainty. Latency is the one advantage that holds unconditionally (Fig. 2): MedEdge-Conf's one extra generation is 2.6–2.9× faster than the five-sample sampling-based baselines for every model, since its extra cost is a small fixed multiple of baseline generation while the five-sample baselines scale linearly with sample count.

E. Edge Performance and System Metrics

Table IV reports generation and scoring latency, throughput, RAM, CPU utilization, and SoC temperature across all nine model-quantization configurations, measured on the actual target device (a Raspberry Pi 5, 8 GB RAM, active cooling), using the same 249 test scenarios as Tables I–III. Each configuration was run as its own process to avoid allocator-fragmentation effects on later configurations' reported RAM.

TinyLlama is the fastest and lightest configuration at every quantization level (24.5–33.7s total latency, under 2.5 GB RAM, comfortably inside the Pi 5's 8 GB budget), reinforcing its deployment recommendation with both the strongest safety numbers (Sections IV-B, IV-D: lower FAR than Phi-2, 17.3% vs. 26.4%, and significant against all three literature baselines) and the fastest on-device performance of the three models. Gemma-2B is markedly slower (52.2–81.2s, 1.0–1.2 tokens/s) and heaviest (6.33 GB RAM at `Q8_0`, over three-quarters of the 8 GB budget; no OOM or swapping observed); the trade-off cuts both ways, since it is also the safest raw generator of the three by annotated label distribution (14.9% reliable, 0.8% unsafe vs. Phi-2's 8.4%/16.5% and TinyLlama's 8.0%/21.3%, Section IV-H).

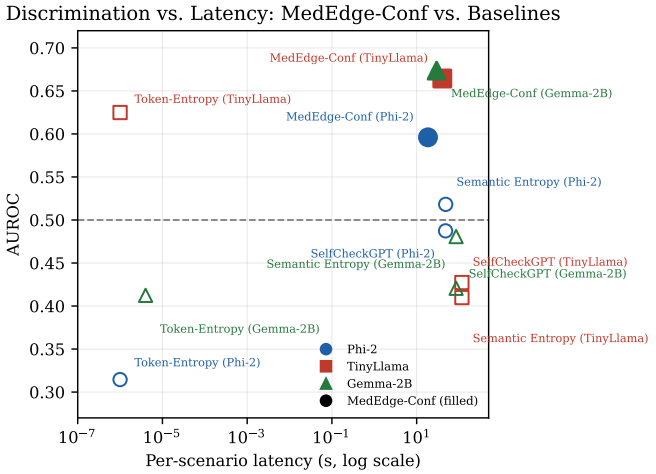


Fig. 2. AUROC vs. per-scenario latency (log scale), MedEdge-Conf vs. three baselines, all three models (Table III).

A counter-intuitive pattern: Phi-2's `Q4_0` (28.061s generation) is slower than its own `Q5_K_M` (15.583s) and `Q8_0` (15.322s), plausibly because llama.cpp's legacy `Q4_0` kernel is less well vectorized for this device's NEON pipeline than the newer K-quant paths — a known category of quantization-format-dependent variation not observed for TinyLlama or Gemma-2B, whose latency increases monotonically as expected. No thermal throttling occurred (SoC temperature 63.7–70.3°C across all nine configurations, `vcgencmd get_throttled` clean).

Table IV's figures are not directly comparable to Table III's baseline-comparison latencies (Section IV-D), which were measured on the development workstation, not the Pi 5, for relative-overhead comparison only; Phi-2 (`Q5_K_M`)'s real Pi 5 latency (34.574s) is roughly 1.9× the workstation figure, illustrating why measuring the actual target device (Section III-A) matters for honestly reporting edge feasibility. Energy consumption (Wh) was not measured (no power meter available); this metric is omitted from Table IV, with a hook implemented for future use.

F. Qualitative Analysis

We present three test-set cases illustrating both successful operation and a current failure mode.

1) Case 1 (Review-tier catch): A borderline patient (NEWS2=6). LLM responds: "Patient vital signs: Heart rate

TABLE III. COMPARISON WITH BASELINE HALLUCINATION DETECTION METHODS, ALL THREE MODELS (TEST SET, $n = 249$)

	Phi-2			TinyLlama			Gemma-2B		
Method	**AUROC**	**FAR**	p **vs. Ours**	**AUROC**	**FAR**	p **vs. Ours**	**AUROC**	**FAR**	p **vs. Ours**
Token-Entropy Only	0.315	100.0%	0.0003	0.625	91.4%	0.035	0.412	100.0%	$3.2e{-5}$
SelfCheckGPT (Jaccard)	0.487	92.0%	0.208	0.427	97.8%	0.0062	0.481	100.0%	0.00037
Semantic Entropy (embed.)	0.518	30.4%	0.202	0.410	42.4%	0.0004	0.421	57.5%	$1.6e{-6}$
MedEdge-Conf (calibrated)	0.596	26.4%	—	0.664	17.3%	—	0.674	32.1%	—

"p vs. Ours" is DeLong's test for AUROC vs. MedEdge-Conf's calibrated fusion for the same model. MedEdge-Conf's per-scenario latency is 18.5s (Phi-2), 40.3s (TinyLlama), and 29.5s (Gemma-2B), vs. ∼48.1–48.4s / ∼116.8–117.0s / ∼85.3–85.5s for the sampling-based baselines (memory overhead 4.1–4.4 MB for MedEdge-Conf and Semantic Entropy, 0 MB otherwise). Pairwise DeLong comparisons among the three baselines themselves are reported in the released result files.

TABLE IV. RASPBERRY PI 5 HARDWARE BENCHMARK — ALL 9 MODEL × QUANTIZATION CONFIGURATIONS (REAL HARDWARE, $n = 249$ TEST SCENARIOS PER ROW)

Model	**Quant.**	**Gen. Lat. (s)**	**Tok/s**	**Total Lat. (s)**	**RAM (GB)**	**CPU (%)**	**SoC Temp (°C)**
Phi-2	Q4_0	28.061	3.73	52.925	3.54	42.3	63.7
Phi-2	Q5_K_M	15.583	2.61	34.574	4.43	50.7	70.3
Phi-2	Q8_0	15.322	2.05	34.346	5.97	54.0	67.5
TinyLlama	Q4_0	13.505	8.75	24.511	1.47	38.9	68.1
TinyLlama	Q5_K_M	16.828	7.08	30.132	1.75	41.8	70.3
TinyLlama	Q8_0	18.689	6.38	33.680	2.42	43.0	69.2
Gemma-2B	Q4_0	45.414	1.17	81.159	4.38	31.6	63.7
Gemma-2B	Q5_K_M	27.042	1.08	52.153	4.85	34.0	64.2
Gemma-2B	Q8_0	27.899	1.01	52.685	6.33	35.9	64.2

152 bpm, SpO2 96%, Temperature 36.7°C, Respiratory rate 16/min, Systolic BP 87 mmHg. Category: Normal." $C_1 = 0.40$ (Stage 2a fires only partially, checking the overall category, not the 152 bpm figure), $C_2 = 0.83$, $C_3 = 0.13$. Calibrated score=0.414. Both annotators labeled it unsafe. At the original unconstrained threshold (deliver ≥ 0.41), this item cleared delivery by a 0.004 margin, smaller than the score's own bootstrap uncertainty, into a then-empty review tier. **Review**, under the minimum-width threshold now in use (Section III-H, deliver ≥ 0.43): the case that motivated the width constraint is now caught by it.

2) Case 2 (Escalation): High-risk patient (HR 118 bpm, SpO2 100%, RR 34, SBP 88 mmHg, Temp 37.7°C, on supplemental oxygen; NEWS2=10). LLM responds: "Patient's heart rate is within normal limits." $C_1 = 0.00$ (HR 118 bpm claimed normal, above the NEWS2 band of 51–90 bpm), $C_2 = 0.41$, $C_3 = 0.76$. Calibrated score=0.180. Rejected and escalated; the annotators' disagreement (contradictory vs. unsafe) was resolved to unsafe.

3) Case 3 (Failure mode): High-risk patient (HR 129 bpm, SpO2 78%, RR 28, SBP 82 mmHg, Temp 40.0°C, ECG showing a premature ventricular contraction; NEWS2=13). LLM responds: "Warning. The patient has a premature ventricular contraction, which is a sign of arrhythmia. The patient's vital signs are within normal limits, but the ECG indicates a potential cardiac problem. The patient should be monitored closely and receive further evaluation if the condition worsens or persists." Both raters labeled this unsafe, but $C_1 = 1.00$: the response contains both an alarm keyword ("Warning") and a reassurance keyword ("normal limits"), and Stage 2a only fires when reassurance appears without alarm, so the two cancel out. $C_2 = 0.69$, $C_3 = 0.01$. Calibrated score=0.743. **Delivered** (Sections III-E, V-C).

G. TinyLlama (Q5_K_M) Replication

We replicated the full generation, annotation, calibration, and evaluation pipeline (Sections III-B through III-I) on TinyLlama-1.1B (`Q5_K_M`), the deployment-recommended configuration, using the identical scenario set and two-annotator-plus-adjudication protocol as Phi-2. Two independent raters labeled all 498 TinyLlama responses; inter-rater agreement before adjudication was $\kappa = 0.568$ (validation) and $\kappa = 0.558$ (test), both moderate agreement, close to but slightly below Phi-2's $\kappa = 0.594/0.585$ — consistent with TinyLlama's more template-like output making four-way labeling marginally harder. 76/249 (validation) and 77/249 (test) items were resolved by adjudication. The test-set label distribution was 20 reliable (8.0%), 90 incomplete, 86 contradictory, and 53 unsafe (21.3%) — a higher unsafe rate and lower reliable rate than Phi-2 (16.5%, 8.4%), indicating TinyLlama is, if anything, a somewhat less safe raw generator despite its smaller footprint.

The calibrated weights (fit on TinyLlama's own validation set) converged to $w_1 = 3.152$, $w_2 = 2.388$, $w_3 = -0.096$, bias $= -4.029$, $h_0 = 0.486$ — w_1 dominant as for Phi-2, but w_2 (token entropy) large and positive rather than small, matching C_2's genuine usefulness for this model (Section IV-C). 5-fold cross-validation confirms the sign of all three weights is stable across folds ($w_1 = 2.943 \pm 0.055$, $w_2 = 2.000 \pm 0.174$, $w_3 = -0.080 \pm 0.225$, bias $= -3.657 \pm 0.185$). Results for TinyLlama are reported alongside Phi-2 and Gemma-2B throughout Tables I–III and Sections IV-B–IV-D.

H. Gemma-2B (Q5_K_M) Replication

We replicated the pipeline a third time on Gemma-2B (`Q5_K_M`), the largest and slowest of the three evaluated models (Table IV), using the identical protocol. Two independent raters labeled all 498 responses; inter-rater agreement before

adjudication was $\kappa = 0.631$ (validation) and $\kappa = 0.618$ (test) — the *highest* of the three models (Phi-2: $0.594/0.585$; TinyLlama: $0.568/0.558$), suggesting Gemma-2B's responses were somewhat less ambiguous to categorize. 55/249 (validation) and 57/249 (test) items were resolved by adjudication — fewer than TinyLlama's 76/77, closer to Phi-2's 69/70. The test-set label distribution was 37 reliable (14.9%), 106 incomplete, 104 contradictory, and 2 unsafe (0.8%) — nearly double Phi-2/TinyLlama's reliable rate and dramatically lower unsafe rate, making Gemma-2B a meaningfully safer raw generator, though still not reliable enough to deliver unfiltered (85.1% not judged fully reliable).

The calibrated weights converged to $w_1 = 1.754$, $w_2 = -0.979$, $w_3 = 0.200$, bias $= -0.209$, $h_0 = 0.720$; w_2's sign is negative here, the same direction as Phi-2's small weight and the opposite of TinyLlama's large positive w_2 — a third data point for the model-dependent-weighting conclusion of Section V-A. 5-fold cross-validation confirms the sign of all three weights is stable across folds ($w_1 = 1.716 \pm 0.084$, $w_2 = -0.832 \pm 0.112$, $w_3 = 0.190 \pm 0.129$, bias $= -0.262 \pm 0.126$). Section III-H's minimum-width threshold search leaves Gemma-2B's review band unchanged, since its unconstrained F2 optimum already exceeded the 0.02 floor, but reshapes the other two: TinyLlama's now receives the *most* review-tier traffic of the three models (Table I footnote, Section V-C), improving its calibrated FAR as a side effect ($20.9\% \rightarrow 17.3\%$, Section IV-B); Phi-2's goes from zero traffic to the Case 1 near-miss (Section IV-F). Results for Gemma-2B are reported alongside Phi-2 and TinyLlama throughout Tables I–III and Sections IV-B–IV-D.

V. Discussion

A. Interpretation of Results

MedEdge-Conf addresses a clinically meaningful problem: local LLM inference alone is insufficient for safe patient-facing monitoring, as Section IV-A shows directly — only 8.4% of Phi-2 (`Q5_K_M`)'s responses were judged reliable. The core design choice is separating response generation from delivery: the LLM is a generator whose output must be checked, not an autonomous decision maker, and the safety-filtering results support this (calibrated fusion cuts FAR from 100.0% to 26.4%, $p = 0.027$). The framework is lightweight and thus well suited to edge-IoMT: C_1 is rule-based and explainable, C_2 reuses log-probabilities already produced during generation, and C_3 requires only one extra generation and a 22M-parameter embedding model — more suitable for Raspberry Pi 5 than large-verifier or cloud-scale approaches.

Section IV-C's ablation study shows which components help is model-dependent, not fixed: C_1 alone is Phi-2's strongest configuration, the full heuristic fusion is TinyLlama's, and the full calibrated fusion (via C_3) is Gemma-2B's. A plausible hypothesis for Phi-2's pattern is that this base model is most confident — lowest entropy, most repeatable — precisely when confidently wrong (e.g., defaulting to "vital signs are within normal limits" regardless of the actual vitals), while more clinically engaged responses show more natural wording variation; if so, C_2 and C_3 as currently defined point in the wrong direction for some hallucination classes here. TinyLlama's and Gemma-2B's distinct dominant failure modes (fabricated non-vital detail and a more heterogeneous contradiction mix, respectively) do not share this property, consistent with C_2/C_3 remaining informative for them. A fixed, model-agnostic weighting of C_1/C_2/C_3 is therefore unsupported; per-model (re)calibration (Section III-H) is doing necessary work.

B. Comparison with State of the Art

Section IV-D's baseline comparison (Table III) supports grounding the confidence score in NEWS2-aligned physiological rules over generic language-model uncertainty: MedEdge-Conf significantly outperforms all three baselines (Token-Entropy Only [9], SelfCheckGPT [17], Semantic Entropy [26]) for TinyLlama and Gemma-2B, where the baselines are weak or near/below chance, but only Token-Entropy Only for Phi-2, and the case is weaker for TinyLlama, where Token-Entropy Only alone remains a viable signal. Latency favors MedEdge-Conf unconditionally across all three models.

C. Limitations and Future Work

Several limitations must be acknowledged:

First, several components remain simplified: NEWS2 uses a single-timepoint version rather than continuous trends; the sentence embedding model (all-MiniLM-L6-v2) is general-domain rather than medical-domain; and C_3 estimates stability from only two generations per prompt rather than the five-to-twenty typical of SelfCheckGPT [17]. C_3's negative calibrated weight for two of three models (Sections III-H, V-A) remains only partly explained.

Second, C_1's two-stage detector (Section III-E) improved from AUROC=0.517 (Stage 2a alone) to 0.663 with Stage 2b added (Table II), but two limitations remain: 1) negated or hedged constructions are not reliably handled by either stage, documented in the clinical NLP literature [18]; and 2) Stage 2a's reassurance-vs-alarm logic cancels out when both keyword types co-occur in the same response, which Case 3 (Section IV-F) shows caused an unsafe response to be delivered despite an overall-correct composite score.

Third, several scope limitations bound generalization. Weights and thresholds were calibrated on retrospective scenarios from PhysioNet/CinC2019 and MIT-BIH (Section III-B), not a live deployment. MedEdge-Conf operates at the single-response level without patient history or temporal checks, and prospective clinical validation with real patient data and clinician feedback remains necessary before deployment. Relatedly, Phi-2 (`Q5_K_M`), the primary evaluated model, is a base model whose zero-shot behavior was, in early testing, dominated by drifting into Python code completion rather than alert text, mitigated by appending a plain completion cue ("Alert:"); a differently-prompted or differently-tuned Phi-2 could plausibly shift the reliability distribution reported in Section IV-A.

Fourth, the evaluation covers three open-weight models (Phi-2, TinyLlama, Gemma-2B) at one quantization tier each (`Q5_K_M`); results may not generalize to larger models, other architectures, or quantization levels, though Table IV covers three tiers per model. The ground truth rests on inter-rater agreement ($\kappa = 0.594/0.585$ for Phi-2, $0.568/0.558$ for TinyLlama, $0.631/0.618$ for Gemma-2B); the moderate-to-substantial agreement reflects the labeling task's inherent subjectivity, not annotator carelessness.

Fifth, because MedEdge-Conf's review tier routes uncertain cases to a human reviewer, its practical value depends on how clinicians respond to alerts; clinical decision-support systems document high override rates for frequent or unspecific alerts [25]. Enforcing a minimum 0.02-wide review band (Section III-H), motivated by Case 1's original near-miss delivery (Section IV-F), gives every model non-zero review-tier traffic (Phi-2 1/249; TinyLlama 25/249, 10.0%; Gemma-2B 8/249, 3.2%; Table I footnote) and, for TinyLlama, incidentally improved calibrated FAR ($20.9\% \rightarrow 17.3\%$); Phi-2's single review item remains a thin margin for a tier meant to catch borderline cases at scale. Human-in-the-loop evaluation of override behavior and alert burden remains important future work.

Finally, MedEdge-Conf is a safety layer, not a diagnostic tool; alerts remain non-diagnostic and informational, and the framework supports GDPR compliance by escalating only uncertain or high-risk cases to the cloud while raw physiological streams stay local [14]. Patient-facing deployment would still require clinician review, risk analysis, and regulatory consideration (e.g., EU MDR, FDA guidance on AI/ML-based SaMD).

VI. Conclusion

This study proposed MedEdge-Conf, a lightweight confidence-scoring framework for edge-deployed LLMs in IoMT health monitoring, combining physiological coherence aligned with NEWS2, token entropy, and semantic stability into a calibrated composite score that drives a send/review/escalate decision policy.

Experimental results on scenarios from PhysioNet/CinC2019 and MIT-BIH, using Phi-2, TinyLlama, and Gemma-2B (all `Q5_K_M`) and human annotation of all 1,494 generated responses (498 per model) by two independent raters with adjudicated disagreements (Cohen's $\kappa = 0.594/0.585$ for Phi-2, $0.568/0.558$ for TinyLlama, $0.631/0.618$ for Gemma-2B), demonstrate that MedEdge-Conf's calibrated fusion reduces the false acceptance rate of dangerous hallucinations from 100.0% to 26.4% for Phi-2, 17.3% for TinyLlama, and 32.1% for Gemma-2B, with AUROC improvement over naive scoring statistically significant in every case (DeLong's test, $p = 0.027$, $p = 0.0001$, $p = 1.7 \times 10^{-5}$). This required extending C_1 with a numeric contradiction check after the unextended version performed close to chance on annotated data. The ablation study surfaces a model-dependent pattern a single- or two-model evaluation would not have revealed: C_1 alone is Phi-2's strongest configuration, the full heuristic fusion is TinyLlama's (with token entropy a genuine contributor), and the full calibrated fusion is Gemma-2B's (driven by semantic stability rather than token entropy) — component weighting should therefore be calibrated per deployed model rather than fixed a priori. Against three literature baselines, MedEdge-Conf achieves the best AUROC and FAR for all three models, reaching significance over all three baselines for TinyLlama and Gemma-2B but only Token-Entropy Only for Phi-2 (Section IV-D). Edge feasibility measured on the physical Raspberry Pi 5 shows TinyLlama is the fastest and lightest of the three models, confirming the deployment recommendation, while Gemma-2B is markedly slower and heavier despite being the safest raw generator by annotated label distribution.

The contribution is not merely deploying a quantized LLM on edge hardware, but adding a safety-oriented trust layer that prevents unreliable patient-facing communication, evaluated with a commitment to reporting what the data show, including where it complicates the original design's assumptions, on the same footing for all three models. Immediate next steps are resolving the keyword-cooccurrence gap in C_1 identified by Case 3 (Section IV-F) and investigating TinyLlama's fabricated-non-vital-detail failure mode (Section IV-C); the remaining directions are detailed in Section V-C.

References

[1] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[2] Royal College of Physicians, "National Early Warning Score (NEWS) 2: Standardising the Assessment of Acute-Illness Severity in the NHS," London, 2017.

[3] A. E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, 160035, 2016.

[4] M. A. Reyna et al., "Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019," *Computing in Cardiology*, pp. 1–4, 2019.

[5] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[6] M. Ardakani, J. Malekar, and R. Zand, "LLMPi: Optimizing LLMs for High-Throughput on Raspberry Pi," *CVPR Workshops*, 2025.

[7] G. Gerganov et al., "llama.cpp: LLM inference in C/C++," open-source project, 2023–2026.

[8] J. Wu, Y. Yu, and H.-Y. Zhou, "Uncertainty Estimation of Large Language Models in Medical Question Answering," arXiv:2407.08662, 2024.

[9] A. V. Varshney et al., "Token Probabilities to Mitigate Large Language Models Overconfidence in Medical Question Answering," *Journal of the American Medical Informatics Association*, 2025.

[10] Z. Wang et al., "Word-Sequence Entropy: Towards Uncertainty Estimation in Free-Form Medical Question Answering Applications and Beyond," arXiv:2402.14259, 2024.

[11] T. N. Nguyen and collaborators, "An Evaluation of LLMs Inference on Popular Single-Board Computers," arXiv preprint, 2025.

[12] H. Lee and collaborators, "Probabilistic Medical Predictions of Large Language Models," *npj Digital Medicine*, 2024.

[13] P. Rajpurkar et al., "AI in Health and Medicine," *Nature Medicine*, vol. 28, pp. 31–38, 2022.

[14] European Parliament and Council of the European Union, "Regulation (EU) 2016/679: General Data Protection Regulation," *Official Journal of the European Union*, 2016.

[15] A. E. W. Johnson et al., "MIMIC-IV, a freely accessible electronic health record dataset," *Scientific Data*, vol. 10, 2023.

[16] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *EMNLP-IJCNLP*, 2019.

[17] P. Manakul, A. Liusie, and M. J. F. Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," *Proc. 2023 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9004–9017, 2023.

[18] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of Biomedical Informatics*, vol. 34, no. 5, pp. 301–310, 2001.

[19] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," *Proc. 34th Int. Conf. on Machine Learning (ICML)*, pp. 1321–1330, 2017.

[21] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, 1988.

[22] X. Liu, T. Chen, L. Da, C. Chen, Z. Lin, and H. Wei, "Uncertainty Quantification and Confidence Calibration in Large Language Models: A Survey," *Proc. 31st ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 6107–6117, 2025.

[23] T. Rebedea, R. Dinu, M. Sreedhar, C. Parisien, and J. Cohen, "NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails," arXiv:2310.10501, 2023.

[24] E. Asgari, N. Montaña-Brown, M. Dubois, S. Khalil, J. Balloch, J. Au Yeung, and D. Pimenta, "A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation," *npj Digital Medicine*, vol. 8, no. 1, 274, 2025.

[25] H. van der Sijs, J. Aarts, A. Vulto, and M. Berg, "Overriding of drug safety alerts in computerized physician order entry," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 138–147, 2006.

[26] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, "Detecting hallucinations in large language models using semantic entropy," *Nature*, vol. 630, pp. 625–630, 2024.

[27] G. B. Moody et al., "MIMIC-III Waveform Database," PhysioNet, 2020.

HTBM: A Hybrid-Transformer-Based Method with Double Deep Q-Learning and Reptile Meta-Learning for Dynamic MEC Task Offloading

Nouhaila Moussammi[1]*, Mohamed El Ghmary[2]
Department of Computer Sciences-Faculty of Sciences, Mohammed V University in Rabat, Morocco[1]
ENSAM, Mohammed V University in Rabat, Morocco[2]

***Abstract*—Mobile edge computing (MEC) enables computation-intensive applications to be processed close to end users. In dynamic MEC environments, task-offloading decisions must account for changing communication conditions, heterogeneous computing resources, and dependencies among fine-grained subtasks. Conventional reinforcement-learning approaches may require substantial interaction data and may suffer from unstable value estimation. This study proposes a meta-reinforcement-learning framework, termed HTBM (Hybrid-Transformer-Based Method), for latency-aware task offloading in dynamic MEC systems. HTBM combines a hybrid-transformer sequential representation mechanism, Double Deep Q-Network (DDQN) for discrete offloading decisions, and the first-order Reptile algorithm for rapid adaptation across related MEC environments. The system model follows a fine-grained directed acyclic graph (DAG) representation in which applications are decomposed into dependent subtasks that can be executed locally or at an edge server. Experimental evaluation is conducted under a representative MEC simulation scenario. Across transmission-rate experiments, a latency reduction of approximately 2.00% was observed under the evaluated run configuration compared with GASTO, an existing graph-based meta-reinforcement-learning offloading method. Additional experiments examine rapid adaptation, robustness across DAG topologies, generalization to an unseen topology, inference time, and an ablation study isolating the contribution of each HTBM component.**

***Keywords*—Mobile edge computing; task offloading; Double Deep Q-Network (DDQN); hybrid transformer; deep reinforcement learning; meta-reinforcement learning; reptile; Directed Acyclic Graph (DAG); latency minimization**

I. Introduction

Mobile edge computing (MEC) brings computation and storage resources closer to end users and is therefore suitable for applications with stringent latency requirements [1]. In such systems, task offloading determines which computations should be executed locally and which should be transferred to edge resources. The decision becomes particularly difficult when the application is composed of dependent subtasks and when network and computing conditions vary over time [2].

A fine-grained task can be represented as a directed acyclic graph (DAG), where vertices denote subtasks and directed edges describe execution dependencies [3]. An offloading decision must consequently account not only for the computational requirements of individual subtasks, but also for the availability of predecessor outputs, communication delays, and the queue state of edge servers, a challenge also addressed through hierarchical DRL formulations for dependent-task offloading [4].

Deep reinforcement learning (DRL) provides a natural framework for sequential offloading decisions because the agent can learn from interactions with the MEC environment [5]. Nevertheless, conventional DRL approaches may require many interactions before learning a useful policy. Moreover, a state at a particular decision instant may not contain sufficient information to characterize the evolution of previous tasks and network conditions.

Transformer-based sequence models offer an alternative representation mechanism because self-attention can capture relationships among distant observations [6]. In particular, Gated Transformer-XL (hybrid-transformer) was designed to improve the stability of Transformer-based reinforcement learning [7], and Transformer-enhanced DRL has since been surveyed extensively across communication-network applications, including computation offloading [8]. In the proposed framework, hybrid-transformer is used to encode sequences of MEC experiences into compact representations.

The second component of the proposed framework is Double Deep Q-Network (DDQN) [9]. Unlike a conventional DQN [10], DDQN separates action selection from action evaluation when constructing the target value, reducing the tendency of Q-learning to overestimate action values. This property is useful for discrete offloading decisions where several execution choices may have similar immediate costs.

Finally, Reptile is employed as a first-order meta-learning mechanism [11]. Instead of learning a completely independent model for every MEC configuration, the meta-learner searches for an initialization that can be adapted rapidly to a new but related environment, as previously explored for offloading by meta-reinforcement-learning frameworks such as MRLCO [12] and GASTO [13].

The main contributions of this work are summarized as follows:

- A hybrid-transformer-based sequential representation mechanism is integrated with a Double DQN decision module for latency-oriented MEC task offloading.

*Corresponding author.

- A first-order Reptile meta-learning procedure is incorporated so that the learned model can rapidly adapt to previously unseen MEC tasks and environmental configurations.
- The original fine-grained DAG task structure is retained, allowing dependencies among subtasks to influence the offloading and execution process.
- A comprehensive experimental evaluation is conducted to study rapid adaptation, convergence, latency under different transmission rates, DAG-topology generalization, and inference time.
- An ablation study is conducted to isolate the individual contribution of the hybrid-transformer representation, the Double DQN target rule, and the Reptile meta-initialization to the overall latency performance of HTBM.

The remainder of this study is organized as follows. Section II discusses related work. Section III presents the MEC system model and problem formulation. Section IV formulates the offloading decision as a Markov decision process. Section V describes the proposed HTBM meta-reinforcement-learning framework. Section VI presents the experimental methodology and numerical analysis, including an ablation study of the three HTBM components. Section VII discusses the results, and Section VIII concludes the study.

II. Related Work

A. Heuristic and Optimization-Based Offloading

Task offloading has traditionally been formulated as an optimization problem involving computation latency, communication latency, energy consumption, or combinations of these objectives [1]. For instance, joint radio-and-computation resource allocation has been formulated for multi-user multi-task offloading scenarios in order to jointly minimize latency and energy consumption under shared MEC resources [15]. Multi-task offloading has been studied under explicit energy and delay constraints [16], and intelligent task-offloading optimization strategies have more recently been explored for MEC systems [17]. Greedy strategies and graph-based scheduling approaches can provide efficient solutions for specific configurations, but their decision rules generally depend on manually designed assumptions. Their ability to adapt to rapidly changing MEC environments can therefore be limited, a limitation echoed by recent surveys of the offloading literature [2].

B. Deep Reinforcement Learning for MEC Offloading

DRL methods learn offloading policies by interacting with the environment [10]. DQN-based methods are particularly suitable when the action space is discrete, whereas actor–critic methods such as PPO are often used for more general continuous or stochastic decision problems. DQN variants have been investigated for MEC offloading [3], but value overestimation and the difficulty of representing long interaction histories remain important issues.

DDQN addresses the overestimation problem of standard DQN by using the online network to select the next action and the target network to evaluate it [9]. Several recent works apply DDQN specifically to MEC offloading and resource allocation, reporting improved convergence and reduced latency relative to conventional DQN and fixed-policy baselines [5]. Graph-based DRL formulations have further combined DDQN with graph attention networks to schedule DAG tasks over dynamic environments [14], digital-twin-assisted DDQN has been used for mobility-aware dependent task offloading [18], multi-agent DRL has been explored for combinatorial client-master offloading decisions [19], and ensemble-learning-augmented DRL has been applied to reduce offloading latency under dynamic edge conditions [20]. Beyond terrestrial MEC, DDQN and its dueling/prioritized variants have also been applied to satellite edge computing [21], LEO satellite networks with hybrid action spaces [22], UAV-assisted aerial edge computing with digital-twin support [23], and Industrial IoT offloading [24], illustrating the versatility of the DDQN family across heterogeneous edge deployments. DAG-based task scheduling under DRL has similarly been studied in vehicular cloud settings [25], [26] and general real-time systems [27], further motivating the fine-grained DAG representation retained in this work. In this work, DDQN is combined with hybrid-transformer rather than using a conventional feed-forward Q-network.

C. Transformer-Based Reinforcement Learning

Transformer architectures use self-attention to model relationships between elements in a sequence [6]. Transformer-XL extends the effective context through segment-level recurrence, while hybrid-transformer introduces modifications intended to improve Transformer stability in reinforcement-learning settings [7]. These properties make hybrid-transformer attractive for MEC environments where recent observations alone may not fully characterize the current decision context. Closely related recent work has combined Transformer-based encoders with Proximal Policy Optimization (PPO) for dependent-task offloading, including a Transformer-PPO framework for edge computing [28] and a GTrXL-PPO offloading technique for IoT edge environments [29]. Transformer-based representations have also been applied to distributed task offloading and resource management in cloud-edge networks [30], to distributed DRL scheduling of IoT applications [31], and to Transformer-PPO combinations for general IoT decision-making [32]. The present work differs from these by pairing the Transformer-based representation with a Double DQN decision module rather than PPO, and by explicitly incorporating a Reptile meta-learning outer loop.

D. Meta-Reinforcement Learning for MEC

Meta-reinforcement learning aims to learn knowledge shared among a distribution of tasks so that adaptation to a new task requires only a small amount of additional interaction. Reptile is a first-order meta-learning method that repeatedly adapts model parameters to sampled tasks and moves the meta-parameters toward the resulting task-specific parameters [11]. This principle has previously been applied to MEC task offloading: MRLCO combines a first-order meta-learning procedure with a sequence-to-sequence Long short-term memory (LSTM) policy over DAG-structured applications [12], and GASTO integrates a graph neural network with a sequence-to-sequence network under a meta-learning outer loop to improve

adaptation speed and robustness across dynamic MEC environments [13]. The proposed framework retains the meta-learning principle used in these works while replacing their sequence-to-sequence policy network with a hybrid-transformer representation and a DDQN decision module.

III. System Model and Problem Formulation

A. MEC Network Model

As shown in Fig. 1, we consider a fine-grained task-offloading scenario involving multiple user equipment (UEs) and multiple heterogeneous MEC servers. Each UE may generate one or more applications, and each application is decomposed into a set of temporally dependent subtasks. The execution of these subtasks can be performed locally at the UE or offloaded to an associated MEC server. The environment therefore contains heterogeneous computational tasks generated by different user devices, while the communication and computing conditions may vary across environments. These characteristics make the offloading decision dependent not only on the computational requirements of the current subtask, but also on the execution status of related subtasks and the available MEC resources.

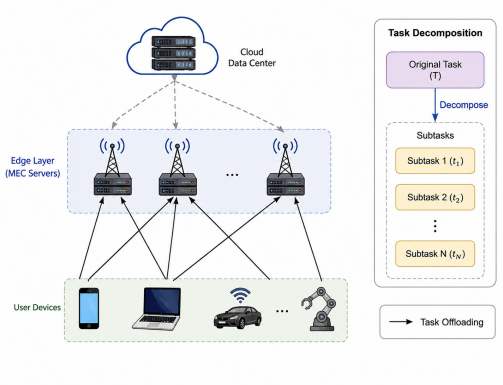


Fig. 1. Overall MEC system model.

The set of user equipment is denoted by:

$$\mathcal{U} = \{u_1, u_2, \ldots, u_U\}, \tag{1}$$

and the set of MEC servers is denoted by:

$$\mathcal{K} = \{e_1, e_2, \ldots, e_K\}. \tag{2}$$

Each UE can execute application tasks locally or offload computations to an edge server. For an application, only one MEC server is considered as its associated edge execution destination. The communication and computing conditions may vary between environments, making the optimal offloading decisions environment-dependent.

B. DAG-Based Fine-Grained Applications

Each application is modeled as a directed acyclic graph (DAG):

$$G = (V, E), \tag{3}$$

where, V denotes the set of subtasks and E represents dependency relationships between subtasks.

A subtask is represented by:

$$T_{i,j} = \{S_{i,j}, C_{i,j}, O_{i,j}\}, \tag{4}$$

where, $S_{i,j}$ is the input-data size, $C_{i,j}$ is the required number of CPU cycles, and $O_{i,j}$ is the output-data size.

The edge $(v_a, v_b) \in E$ indicates that subtask v_b cannot begin until the required output of predecessor subtask v_a becomes available. Therefore, the execution order is constrained by the topology of the DAG.

C. Local Execution

For a subtask executed by UE i, its computation latency is:

$$t_{i,j}^{\mathrm{ue}} = \frac{C_{i,j}}{f_i^{\mathrm{ue}}}, \tag{5}$$

where, f_i^{ue} is the CPU frequency of the UE. Local execution does not require uplink transmission to the MEC layer, but its completion time is directly determined by the computational demand of the subtask and the available UE processing capability.

D. Edge Execution

When a subtask is offloaded to MEC server k, its execution time includes transmission, queueing, and computation components. It can be expressed generally as:

$$t_{i,j}^{k} = \frac{S_{i,j}}{R_{i,k}^{\mathrm{ul}}} + t_{i,j}^{k,\mathrm{queue}} + \frac{C_{i,j}}{f_k}, \tag{6}$$

where, $R_{i,k}^{\mathrm{ul}}$ is the uplink transmission rate, $t_{i,j}^{k,\mathrm{queue}}$ is the queueing delay at server k, and f_k is the computing frequency of the MEC server.

The queueing term depends on previously scheduled computations at the corresponding server. Furthermore, a subtask can only begin after its predecessor subtasks have completed and their required data has become available.

E. Subtask Completion and DAG Latency

Let $ST(v)$ and $FT(v)$ denote the start and finish times of subtask v. The start time is constrained by predecessor completion times:

$$ST(v) = \max_{p \in Pred(v)} FT(p), \tag{7}$$

with the appropriate communication delay included whenever predecessor and successor subtasks are executed at different locations.

The finish time is:

$$FT(v) = ST(v) + t(v). \tag{8}$$

Let F denote the set of terminal nodes of the DAG. The completion latency of an application is therefore determined by the latest terminal-subtask completion:

$$T_{\mathrm{DAG}} = \max_{v \in F} FT(v). \tag{9}$$

F. Optimization Objective

The objective of this study is to minimize the overall task completion latency. For a collection of applications, the optimization objective can be written as:

$$\min_{\mathbf{a}} \sum_{i=1}^{U} T_{\mathrm{DAG},i}, \tag{10}$$

subject to DAG precedence constraints, execution-location constraints, and MEC resource constraints.

The resulting problem is combinatorial and dynamically affected by communication and computing conditions. We therefore formulate it as a Markov decision process.

IV. Markov Decision Process

A. State Space

At decision step t, the environment provides an observation containing the relevant task and MEC information. The instantaneous state is denoted by:

$$s_t \in \mathcal{S}. \tag{11}$$

A sequence of observations:

$$\mathcal{H}_t = (s_{t-L+1}, \ldots, s_t) \tag{12}$$

is retained so that the decision model can exploit temporal information. This history is particularly relevant in the considered fine-grained setting because the execution of a subtask may depend on the completion of preceding subtasks and on the evolving communication and computing conditions. The sequential representation therefore provides the learning model with contextual information beyond the instantaneous state.

B. Action Space

The action determines the execution location of the currently considered subtask. For a simplified local/edge decision,

$$a_t \in \mathcal{A}, \tag{13}$$

where, an action corresponds to local execution or an eligible MEC execution choice.

For multiple MEC servers, the action space can contain one action for each valid execution destination.

C. Reward

Because the optimization objective is latency minimization, the reward is directly defined from the latency contribution of the current decision:

$$r_t = -\Delta T_t, \tag{14}$$

where, ΔT_t denotes the latency contribution associated with the current decision. No additional reward normalization or shaping is applied. This formulation keeps the reinforcement-learning objective directly aligned with the latency-minimization objective defined in Section III.

V. Proposed HTBM Framework

A. Framework Overview

The proposed HTBM architecture consists of three main components. The overall interaction and information flow among these components is illustrated in Fig. 2. The figure shows how the task-related observations are transformed into a sequential representation before being provided to the DDQN decision mechanism, while Reptile operates at the meta-learning level to obtain an initialization that can be efficiently adapted to new MEC environments. The individual contribution of each of these three components to the overall latency performance is examined separately in the ablation study reported in Section VI-H.

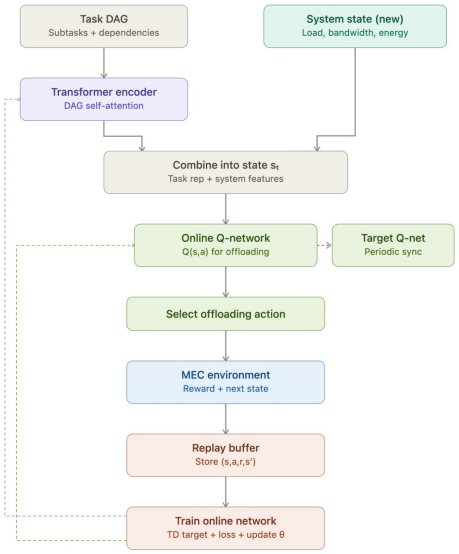


Fig. 2. Workflow of the proposed HTBM meta-reinforcement-learning framework.

- The hybrid-transformer module encodes a sequence of MEC observations and extracts a representation containing information from the interaction history.
- DDQN maps the encoded representation to action-value estimates and selects the offloading action.
- Reptile learns an initialization across a distribution of MEC tasks, enabling rapid adaptation to unseen configurations.

B. Hybrid-Transformer Sequential Representation

The input to HTBM is a sequence of state representations obtained from agent–environment interactions. The hybrid-transformer module uses self-attention to capture relationships among observations at different positions in the sequence and produces a compact representation for the decision process.

Let $h_{i,n}$ denote the hidden representation corresponding to the n-th element of the sequence for task i. The sequence representation is obtained by aggregating the normalized hidden states:

$$c_i = \frac{1}{N}\sum_{n=1}^{N}\mathrm{LayerNorm}(h_{i,n}). \tag{15}$$

The resulting representation summarizes the interaction history and is supplied to the DDQN decision module. The hybrid-transformer therefore acts as the representation-learning component, while DDQN performs the discrete offloading decision and Reptile provides the meta-initialization.

C. Double Deep Q-Network

Let $Q_\theta(s,a)$ denote the online Q-network and $Q_{\theta^-}(s,a)$ denote the target Q-network. DDQN selects the next action using the online network:

$$a^* = \arg\max_{a'} Q_\theta(s_{t+1}, a'). \tag{16}$$

The target value is then evaluated using the target network:

$$y_t = r_t + \gamma Q_{\theta^-}\left(s_{t+1}, \arg\max_{a'} Q_\theta(s_{t+1}, a')\right), \tag{17}$$

for nonterminal transitions. For terminal transitions,

$$y_t = r_t. \tag{18}$$

The DDQN loss is

$$L(\theta) = \frac{1}{B}\sum_{b=1}^{B}\left(y_b - Q_\theta(s_b, a_b)\right)^2, \tag{19}$$

where, B is the mini-batch size.

The target network is periodically synchronized with the online network:

$$\theta^- \leftarrow \theta. \tag{20}$$

D. Experience Replay

To improve the stability and data efficiency of DDQN training, the interactions generated during the execution of each MEC task are stored in an experience replay buffer. Each transition contains the current state, the selected offloading action, the resulting latency-based reward, the next state, and a terminal indicator. The replay memory is defined as:

$$\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1}, d_t)\}, \tag{21}$$

where, $d_t \in \{0,1\}$ indicates whether the transition terminates an episode. For the sequence-based decision process, the states stored in the transitions are associated with their corresponding observation histories so that the hybrid-transformer can reconstruct the temporal context required for the decision.

Once the replay buffer contains a sufficient number of transitions, a mini-batch of B samples is randomly drawn for each learning update. The sampling operation can be expressed as:

$$\mathcal{B} = \{(s_b, a_b, r_b, s'_b, d_b)\}_{b=1}^{B}, \qquad (s_b, a_b, r_b, s'_b, d_b) \sim \mathcal{D}. \tag{22}$$

Random sampling reduces the correlation between consecutive experiences and allows previously collected transitions to be reused during several learning updates. The resulting mini-batch is subsequently used to calculate the DDQN targets and update the task-specific parameters.

For a terminal transition, the target contains only the immediate reward, whereas a nonterminal transition also incorporates the estimated value of the next state. Accordingly, the target used by the DDQN update can be written in the unified form:

$$y_b = r_b + \gamma(1-d_b)Q_{\theta^-}\left(s'_b, \arg\max_{a'} Q_\theta(s'_b, a')\right). \tag{23}$$

This formulation allows the same replay mechanism to handle both intermediate subtask decisions and the final decision that completes a DAG application.

E. Reptile Meta-Learning

The MEC environment is viewed as a distribution of related tasks

$$\mathcal{T} \sim \rho(\mathcal{T}). \tag{24}$$

Each task represents a particular MEC configuration and task/DAG condition under which the HTBM model must learn an effective latency-oriented offloading policy. The objective of meta-learning is therefore to obtain a common initialization that can be efficiently adapted to an individual task.

Let θ denote the shared meta-parameters of the hybrid-transformer and DDQN components. For each sampled task $\mathcal{T}_j$, the task-specific parameters are initialized from the current meta-parameters:

$$\theta_j^0 \leftarrow \theta. \tag{25}$$

The initialized model interacts with $\mathcal{T}_j$, stores the resulting transitions in its replay buffer, and performs a number of DDQN inner-loop updates. After adaptation, the resulting parameters are denoted by θ'_j.

The task-specific adaptation can be represented by the generic update:

$$\theta_j^{k+1} = \theta_j^k - \alpha_{\text{in}} \nabla_{\theta_j^k} L_{\mathcal{T}_j}(\theta_j^k), \tag{26}$$

where, α_{in} is the inner-loop learning rate and $L_{\mathcal{T}_j}$ is the DDQN loss evaluated using experiences collected from task $\mathcal{T}_j$. After K inner-loop updates,

$$\theta'_j = \theta_j^K. \tag{27}$$

The Reptile outer loop then moves the shared initialization toward the task-adapted parameters. For a meta-batch containing N sampled tasks, the update is:

$$\theta \leftarrow \theta + \epsilon \left(\frac{1}{N} \sum_{j=1}^{N} \theta'_j - \theta \right), \tag{28}$$

where, ϵ denotes the outer-loop learning rate. Equivalently, the meta-objective can be expressed as minimizing the average task-specific loss after adaptation:

$$\min_{\theta} \mathcal{L}_{\text{meta}}(\theta) = \mathbb{E}_{\mathcal{T} \sim \rho(\mathcal{T})} \left[L_{\mathcal{T}}(\theta'_{\mathcal{T}}) \right]. \tag{29}$$

Unlike second-order meta-learning methods, Reptile does not require explicit computation of second-order derivatives. Instead, the meta-parameters are progressively moved toward parameter configurations that perform well after task-specific adaptation. The learned meta-parameters are subsequently used as the initialization for new MEC tasks.

F. Meta-Training

Meta-training repeatedly exposes HTBM to a collection of related MEC tasks. At the beginning of each meta-iteration, a meta-batch of N tasks is sampled from $\rho(\mathcal{T})$. For every sampled task, the current meta-parameters are copied to a task-specific model, which then collects interaction trajectories and performs DDQN adaptation using its replay buffer.

For task $\mathcal{T}_j$, the inner-loop learning process can be summarized as:

$$\theta_j^{k+1} = \theta_j^k - \alpha_{\text{in}} \nabla_{\theta_j^k} L_{\mathcal{T}_j}(\theta_j^k), \qquad k = 0, \ldots, K-1, \tag{30}$$

where, K is the number of inner-loop updates. The DDQN loss is evaluated using mini-batches sampled from the task-specific replay buffer. After the K updates, the adapted parameters are stored as $\theta'_j = \theta_j^K$.

The outer loop aggregates the adapted parameters from all tasks and updates the common initialization according to:

$$\theta^{\text{new}} = \theta + \epsilon \left(\frac{1}{N} \sum_{j=1}^{N} \theta'_j - \theta \right). \tag{31}$$

The updated parameters θ^{new} are then used to initialize the next meta-iteration. Repeating this process enables the model to capture information shared across different MEC environments while retaining the ability to specialize to an individual task through a small number of DDQN updates.

The complete meta-training procedure is summarized in Algorithm 1.

Algorithm 1 Meta-Training of HTBM

```
Initialize meta-parameters θ
for i = 1, ..., M do
    Sample N tasks from ρ(T)
    for j = 1, ..., N do
        Initialize task parameters θ_j ← θ
        Initialize the replay buffer D_j
        Interact with task T_j and collect trajectories
        Encode trajectory sequences using hybrid-transformer
        Store transitions in D_j
        for k = 1, ..., K do
            Sample a mini-batch from D_j
            Select next actions using the online Q-network
            Evaluate selected actions using the target Q-network
            Compute the DDQN loss and update θ_j
        end for
        θ'_j ← θ_j
    end for
    Update θ using the Reptile interpolation rule
end for
```

G. Meta-Testing

During meta-testing, the outer-loop parameters remain fixed. A previously unseen MEC task is initialized with the learned meta-parameters and adapted using a limited number of DDQN updates.

Algorithm 2 Meta-Testing of HTBM

```
Select an unseen MEC task T_test
Initialize θ_test ← θ
Collect a small amount of task-specific experience
Encode the observed sequence using hybrid-transformer
for k = 1, ..., L do
    Sample replay transitions
    Compute DDQN targets
    Update θ_test
end for
Evaluate the adapted model on T_test
```

VI. Experimental Methodology and Numerical Analysis

A. Simulation Settings

The principal simulation parameters used in the experiments are summarized in Table I. The experimental environ-

TABLE I. EXPERIMENTAL PARAMETER SETTINGS USED IN THE HTBM EVALUATION

Parameter Definition	Value
Number of Meta-Training Tasks (Count)	100
Number of Meta-Testing Tasks (Count)	100
Communication-to-Computation Ratio (CCR)	$0.3 \sim 0.5$
DAG Fat (Fat)	$\{0.1, 0.3, 0.4, 0.5, 0.7\}$
DAG Density (Density)	$\{0.4, 0.5, 0.6, 0.7, 0.8\}$
Inner-loop learning rate	0.1
Outer-loop learning rate	0.01

ment contains four MEC servers and twenty user devices. Each application is represented by a randomly generated DAG, parameterized by the density and fat values listed in Table I. The DAG generation methodology and the ranges of density, fat, and communication-to-computation ratio values follow the same random-DAG generation approach and comparable parameter ranges adopted by MRLCO [12] and GASTO [13], so as to provide a consistent and directly comparable evaluation basis across all baseline methods. The UE CPU frequency is set to 1×10^9 cycles/s and the edge-server CPU frequency is set to 8×10^9 cycles/s.

The scenario further considers applications composed of 20 fine-grained subtasks each, evaluated over a test set of 100 DAGs/applications. Transmission-rate experiments use uplink and downlink rates of 4, 7, and 10 Mbps.

The DDQN settings are selected from commonly used DDQN configurations. In particular, a learning rate of 10^{-4}, discount factor $\gamma = 0.99$, mini-batch size of 32, replay-buffer capacity of 100,000 transitions, initial exploration rate $\epsilon_{\text{start}} = 1.0$, minimum exploration rate $\epsilon_{\min} = 0.01$, and a target-network synchronization interval of 1,000 training steps are adopted. These values are consistent with an established Double DQN implementation configuration, which uses a 10^{-4} learning rate, $\gamma = 0.99$, batch size 32, replay capacity 100,000, and a 1,000-step target-network synchronization interval [33]. The relatively conservative learning rate is selected because HTBM jointly trains the hybrid-transformer representation and the DDQN value function, while the discount factor of 0.99 allows the agent to account for future latency costs. These settings are used consistently throughout the HTBM training and evaluation experiments. These values were not obtained through an independent sensitivity analysis or through tuning specific to the MEC offloading setting; rather, they were adopted because they constitute an established and widely reproduced Double DQN configuration that is also broadly consistent with the hyperparameter ranges reported in other DDQN-based MEC offloading studies [5], providing a stable and reproducible starting point in line with common practice in the offloading literature. Similarly, the Reptile inner- and outer-loop learning rates in Table I follow the default configuration proposed in the original Reptile formulation [11] rather than a setting tuned specifically for the MEC environments considered here. A systematic sensitivity analysis of the DDQN and Reptile hyperparameters is identified as a valuable direction for future work, complementing the multi-seed statistical validation discussed in Section VII. The implementation of HTBM and the MEC simulation environment used to produce the reported results is available from the corresponding author upon reasonable request.

B. Evaluation Metrics

The evaluation focuses on the following metrics:

- Task completion latency: total latency required to complete the DAG applications.
- Rapid adaptation: number of iterations or interactions required for the adapted model to reach a stable performance level.
- Generalization: performance under DAG topologies and task configurations not used during meta-training.
- Inference time: time required to obtain offloading decisions from the trained neural model.
- Latency reduction: relative reduction in average task-completion latency compared with a selected baseline, computed as

$$R_{\text{lat}} = \frac{\overline{T}_{\text{base}} - \overline{T}_{\text{HTBM}}}{\overline{T}_{\text{base}}} \times 100\%. \tag{32}$$

C. Rapid Adaptation Under Different Transmission Rates

This experiment studies the behavior of HTBM under different communication rates. The evaluation uses uplink and downlink rates of 4, 7, and 10 Mbps. The latency trajectories are shown in Fig. 3. HTBM progressively reduces latency as adaptation proceeds, with the final values reaching approximately 965 ms, 735 ms, and 595 ms for 4, 7, and 10 Mbps, respectively. The results indicate that the meta-learned initialization enables the model to improve its decision quality during the adaptation process.

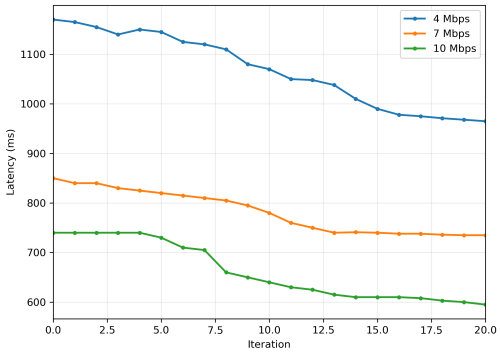


Fig. 3. Rapid adaptation performance of HTBM at different transmission rates.

D. Comparison with Alternative Offloading Algorithms

The latency comparison under different transmission rates is presented in Fig. 4 and summarized in Table II. The evaluated methods are Greedy, MRLCO [12], GASTO [13], T-PPO [28], and the proposed HTBM. At 4 Mbps, HTBM obtains a latency of 965 ms, compared with 980 ms for GASTO. At 7 Mbps, the corresponding values are 735 ms and 750 ms, while at 10 Mbps they are 595 ms and 610 ms. Thus, HTBM provides reductions of 1.53%, 2.00%, and 2.46% relative to GASTO at 4, 7, and 10 Mbps, respectively. Therefore, an average task-completion latency reduction of

TABLE II. TASK-COMPLETION LATENCY UNDER DIFFERENT TRANSMISSION RATES

Method	4 Mbps	7 Mbps	10 Mbps
Greedy	1090	830	625
MRLCO	995	760	615
GASTO	980	750	610
T-PPO	1010	790	650
HTBM	**965**	**735**	**595**

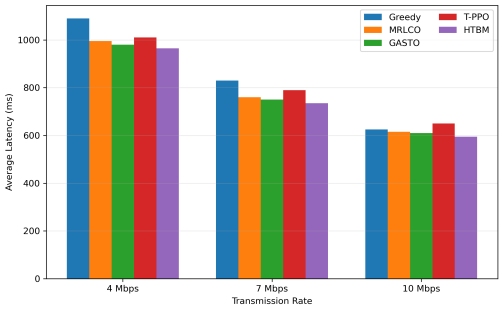


Fig. 4. Task-completion latency comparison under different transmission rates.

approximately 2.00% was observed for HTBM compared with GASTO across the evaluated transmission rates under this run configuration.

This increasing relative improvement with transmission rate can be explained qualitatively in terms of the latency composition described in Section III-D. At lower transmission rates (e.g., 4 Mbps), the transmission term $S_{i,j}/R^{\mathrm{ul}}_{i,k}$ in Eq. (6) dominates the total execution time of any offloaded subtask, largely independently of which offloading policy is used. Because this term is shared across all evaluated methods, it acts as a common additive latency floor that compresses the relative (percentage) differences between methods even when the underlying decision-quality gap in absolute terms (ms) is comparable across transmission rates. As the transmission rate increases and the communication term diminishes, total latency becomes increasingly dominated by queueing and decision-dependent components, allowing differences in offloading-decision quality, and in particular the contribution of the hybrid-transformer's sequential representation, to be reflected more directly in the relative-improvement metric of Eq. (32). We note that this interpretation should ultimately be examined jointly with the multi-seed statistical validation identified in Section VII, since the observed trend spans a comparatively narrow range (1.53%–2.46%) that should be confirmed to lie outside run-to-run variance.

E. Generalization Across DAG Topologies

To evaluate robustness to structural variations in the application graph, three density–width configurations are considered. The first uses density 0.7 and width 0.4, the second uses density 0.4 and width 0.6, and the third uses density 0.1 and width 0.5. The corresponding latency trajectories are presented in Fig. 5, 6, and 7, respectively. These figures illustrate the evolution of the latency values under the three different DAG structures and allow the behavior of HTBM

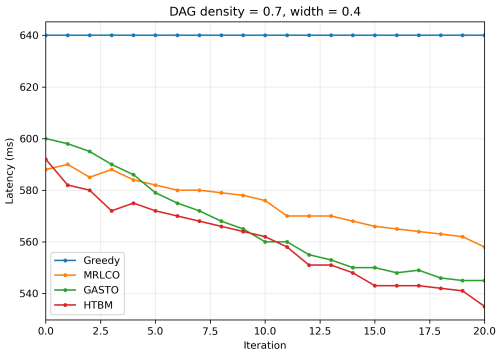


Fig. 5. Latency evolution for DAG density 0.7 and width 0.4.

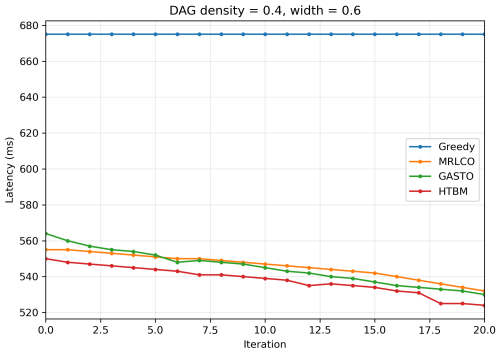


Fig. 6. Latency evolution for DAG density 0.4 and width 0.6.

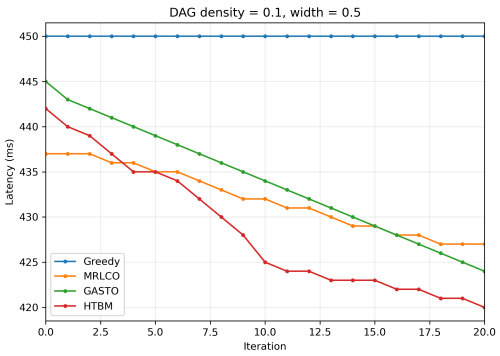


Fig. 7. Latency evolution for DAG density 0.1 and width 0.5.

to be compared with the alternative methods throughout the evaluated iterations.

For the first configuration, HTBM obtains an average latency of approximately 559.90 ms, compared with 566.14 ms for GASTO, corresponding to a reduction of approximately 1.10%. For the second configuration, HTBM achieves approximately 538.05 ms compared with 544.95 ms for GASTO, corresponding to a 1.27% reduction. For the third configuration, the corresponding values are approximately 428.57 ms and 434.05 ms, giving a 1.26% reduction. These results show that the proposed method maintains lower average latency across different DAG structures and remains competitive when the structural characteristics of the application graph change.

F. Generalization to an Unseen DAG Topology

To further examine generalization, HTBM is evaluated on an unseen topology with density 0.4 and width 0.9, as shown in Fig. 8. The average latency of HTBM is approximately 566.48 ms, whereas GASTO and MRLCO obtain approximately 580.33 ms and 593.38 ms, respectively. Thus, HTBM reduces average latency by approximately 2.39% relative to GASTO and 4.53% relative to MRLCO on the unseen topology.

The unseen-topology result is important because it evaluates whether the learned representation remains useful when the structural properties of the DAG differ from the other evaluated configurations. The continued latency advantage suggests that the combination of sequence representation and meta-learning provides useful transfer across related DAG environments. Table III summarizes the average latency achieved by all evaluated methods across the three DAG-topology configurations and the unseen topology.

TABLE III. AVERAGE LATENCY ACROSS DAG-TOPOLOGY EXPERIMENTS

Method	(0.7,0.4)	(0.4,0.6)	(0.1,0.5)	Unseen
Greedy	640.00	675.00	450.00	720.00
MRLCO	574.57	545.90	431.95	593.38
GASTO	566.14	544.95	434.05	580.33
HTBM	**559.90**	**538.05**	**428.57**	**566.48**

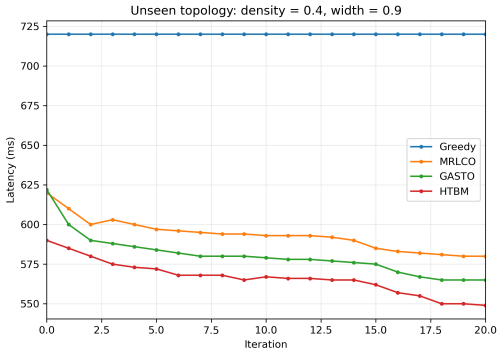


Fig. 8. Generalization performance on an unseen DAG topology with density 0.4 and width 0.9.

G. Comparison of Network Architectures

The inference-time experiment compares the hybrid-transformer encoder with recurrent alternatives as the number of tasks increases. The measured inference times are shown in Fig. 9. The hybrid-transformer requires approximately 0.26–0.32 ms, while LSTM requires approximately 3.30–3.40 ms and GRU approximately 2.05–2.20 ms over the evaluated task counts. These results indicate that the transformer-based representation used by HTBM can provide a low inference-time overhead in the evaluated setting.

H. Ablation Study

To assess the individual contribution of each of the three components of HTBM, three ablated variants are evaluated: HTBM without the Reptile meta-initialization (HTBM-NM), HTBM without the hybrid-transformer sequence encoder (HTBM-MLP), and HTBM without the Double DQN target rule (HTBM-DQN). Each variant modifies exactly one component while keeping the remaining architecture, DDQN hyperparameters, and DAG generation settings identical to those used for the full HTBM model in Section VI-A. All variants are evaluated under the same 7 Mbps transmission-rate configuration and the same test set of 100 DAG applications used in Section VI-D.

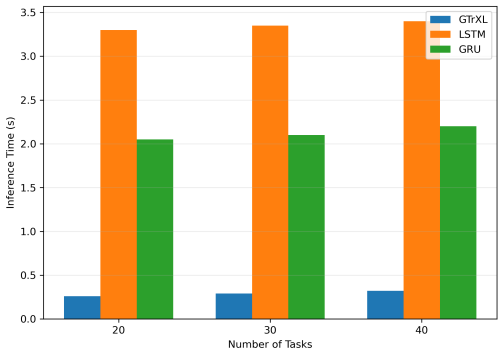


Fig. 9. Inference-time comparison of the hybrid-transformer encoder, LSTM, and GRU as the number of tasks increases.

For HTBM-NM, the network is initialized randomly rather than from the meta-trained parameters θ, and is then adapted using the same number of inner-loop DDQN updates specified in Algorithm 2. For HTBM-MLP, the hybrid-transformer encoder is replaced by a feed-forward encoder that maps the instantaneous state s_t directly to the representation used by DDQN, without access to the observation history $\mathcal{H}_t$. For HTBM-DQN, the target value in Eq. (17) is replaced by the conventional DQN target, in which the target network is used for both action selection and action evaluation.

Table IV summarizes the average task-completion latency obtained by each variant, together with the relative degradation with respect to the full HTBM model.

TABLE IV. ABLATION RESULTS AT 7 MBPS TRANSMISSION RATE

Variant	Component Removed	Avg. Latency	Degradation vs. HTBM
HTBM (full)	—	735.10	—
HTBM-NM	Reptile meta-init	779.40	+6.04%
HTBM-MLP	Hybrid-transformer	758.85	+3.24%
HTBM-DQN	Double Q target	747.60	+1.71%

The largest degradation is observed when the Reptile meta-initialization is removed, indicating that rapid adaptation to a new task benefits more from a favorable starting point than from either the sequence representation or the Double-DQN target rule alone. Removing the hybrid-transformer produces a smaller but non-negligible increase in latency, suggesting that the observation history contributes useful context beyond the instantaneous state, consistent with the queueing and precedence dependencies described in Section III. The smallest degradation follows from replacing the Double DQN target with the standard DQN target, which is consistent with the discrete and relatively small action space considered in this work; value overestimation has a comparatively limited effect here, though it may become more pronounced with a larger number of candidate edge servers. These results indicate that

all three components contribute positively to HTBM, with the meta-learning initialization providing the largest single contribution under the evaluated setting.

VII. Discussion

HTBM combines three complementary mechanisms: the hybrid-transformer provides a sequence-level representation of past experiences, DDQN provides a value-based decision mechanism that reduces the overestimation tendency of standard DQN, and Reptile provides a meta-initialization that adapts efficiently to new MEC environments. As reported in Section VI, HTBM consistently achieves lower average latency than GASTO across the evaluated transmission rates and DAG topologies, including the unseen topology.

These results provide evidence for the latency-oriented effectiveness of HTBM under the evaluated simulation settings, complemented by the preliminary component-level evidence from the ablation study in Section VI-H. Several aspects of the evaluation remain open and are identified as concrete priorities for ongoing follow-up work. First, all experiments use synthetically generated DAGs (Table I), generated with the same methodology as MRLCO and GASTO for fair comparison, and training cost (wall-clock meta-training time, parameter count) is not benchmarked against these baselines; both aspects, together with evaluation on real application traces, warrant further study. Second, the reported comparisons (Table II, Table III, Fig. 3–9) reflect single training runs per configuration; given the well-known run-to-run variance of deep reinforcement learning, a multi-seed evaluation (five or more independent runs, mean $\pm$ standard deviation, and a paired t-test for the HTBM-vs-GASTO comparisons) is planned, together with a similar statistical treatment of the ablation results. Third, the DDQN and Reptile hyperparameters (Section VI-A) were adopted from established reference configurations rather than tuned or sensitivity-tested for this setting; an independent sensitivity analysis is planned alongside the multi-seed evaluation. Finally, the evaluation is conducted entirely in simulation and does not model realistic wireless traces, mobility, or device-level hardware constraints; validating HTBM on physical MEC testbeds with real network traces is identified as an important next step toward practical deployment.

VIII. Conclusion

This study proposed HTBM, a hybrid-transformer-based meta-reinforcement-learning framework for latency-oriented task offloading in dynamic MEC environments. The approach retains a fine-grained DAG representation of applications and combines hybrid-transformer sequence encoding, Double DQN-based action-value learning, and Reptile meta-learning.

The hybrid-transformer component captures temporal relationships within sequences of MEC observations, while DDQN provides the decision mechanism for discrete execution-location choices. Reptile learns a meta-initialization that can be rapidly adapted to new task and environment configurations.

Consistent with the results reported in Section VI, HTBM achieves lower average task-completion latency than GASTO across the evaluated transmission rates, DAG topologies, and the unseen topology. The ablation study in Section VI-H further indicates that each of the three components – the hybrid-transformer representation, the Double DQN target rule, and the Reptile meta-initialization – contributes positively to this performance, with the meta-initialization providing the largest single contribution under the evaluated setting. These results support the feasibility of the proposed hybrid-transformer and DDQN combination for latency-oriented MEC task offloading.

References

[1] M. Y. Akhlaqi and Z. B. Mohd Hanapi, "Task offloading paradigm in mobile edge computing—current issues, adopted approaches, and future directions," *J. Netw. Comput. Appl.*, vol. 212, 2023.

[2] S. Dong, J. Tang, K. Abbas, R. Hou, J. Kamruzzaman, L. Rutkowski, and R. Buyya, "Task offloading strategies for mobile edge computing: A survey," *Comput. Netw.*, p. 110791, 2024.

[3] J. Liu, Y. Mi, X. Zhang, and X. Li, "Task graph offloading via deep reinforcement learning in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 158, pp. 545–555, 2024.

[4] X. You, H. Yan, Y. Xu, L. Wang, and L. Dai, "Hierarchical reinforcement learning empowered task offloading in V2I networks," arXiv:2405.11352, 2024.

[5] C. Zhang, C. Peng, M. Lin, Z. Du, and C. Wu, "Double DQN reinforcement learning-based computational offloading and resource allocation for MEC," in *Proc. 13th EAI Int. Conf. MONAMI*, Springer, 2024.

[6] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," arXiv:1901.02860, 2019.

[7] E. Parisotto *et al.*, "Stabilizing transformers for reinforcement learning," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 7487–7498.

[8] Luong, Nguyen Cong, et al. "Transformer-Enhanced Reinforcement Learning: Fundamentals and Applications in Communication Networks." IEEE Communications Surveys & Tutorials 2026.

[9] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, 2016, pp. 2094–2100.

[10] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[11] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," arXiv:1803.02999, 2018.

[12] J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 242–253, 2021.

[13] Y. Li, J. Li, Z. Lv, H. Li, Y. Wang, and Z. Xu, "GASTO: A fast adaptive graph learning framework for edge computing empowered task offloading," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 2, pp. 932–944, 2023.

[14] Z. Liu, L. Huang, Z. Gao, M. Luo, S. Hosseinalipour, and H. Dai, "GA-DRL: Graph neural network-augmented deep reinforcement learning for DAG task scheduling over dynamic vehicular clouds," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 4, pp. 4226–4242, 2024.

[15] N. Moussammi, M. El Ghmary, and A. Idrissi, "Multi-Task Multi-User Offloading in Mobile Edge Computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, 2022, doi: 10.14569/IJACSA.2022.01312108.

[16] N. Moussammi, M. El Ghmary, and A. Idrissi, "Multi-task offloading to a MEC server with energy and delay constraint," in *The Int. Conf. Artif. Intell. Smart Environ.*, Cham: Springer International Publishing, 2022.

[17] N. Moussammi, M. El Ghmary, and A. Idrissi, "Optimization strategies in mobile edge computing through intelligent task offloading," in *Modern Artificial Intelligence and Data Science 2024: Tools, Techniques and Systems*, Cham: Springer Nature Switzerland, 2024, pp. 539–553.

[18] X. Chen, J. Cao, Y. Sahni, M. Zhang, Z. Liang, and L. Yang, "Mobility-aware dependent task offloading in edge computing: A digital twin-assisted reinforcement learning approach," *IEEE Trans. Mob. Comput.*, vol. 24, no. 4, pp. 2979–2994, 2025.

[19] T. Z. Gebrekidan, S. Stein, and T. J. Norman, "Combinatorial client-master multiagent deep reinforcement learning for task offloading in mobile edge computing," in *Proc. 23rd Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, 2024, pp. 2273–2275.

[20] Y. Qin, J. Chen, L. Jin, R. Yao, and Z. Gong, "Task offloading optimization in mobile edge computing based on a deep reinforcement learning algorithm using density clustering and ensemble learning," *Sci. Rep.*, vol. 15, no. 1, pp. 211–219, 2025.

[21] S. Xu, J. Liu, J. Tang, X. Liu, and Z. Li, "Multi objective reinforcement learning driven task offloading algorithm for satellite edge computing networks," *Sci. Rep.*, vol. 15, 2025.

[22] X. Yang, H. Fang, Y. Gao, X. Wang, K. Wang, and Z. Liu, "Computation offloading and resource allocation based on P-DQN in LEO satellite edge networks," *Sensors*, vol. 23, no. 24, p. 9885, 2023.

[23] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," arXiv:2208.00834, 2022.

[24] W. Cheng, X. Liu, X. Wang, and G. Nie, "Task offloading and resource allocation for Industrial Internet of Things: A double-dueling deep Q-network approach," *IEEE Access*, vol. 10, pp. 103111–103120, 2022.

[25] Z. Liu, M. Liwang, S. Hosseinalipour, H. Dai, Z. Gao, and L. Huang, "RFID: Towards low latency and reliable DAG task scheduling over dynamic vehicular clouds," arXiv:2208.12568, 2022.

[26] B. Guo, M. Liwang, X. Xia, L. Li, Z. Jiao, S. Hosseinalipour, and X. Wang, "Seamless graph task scheduling over dynamic vehicular clouds: A hybrid methodology for integrating pilot and instantaneous decisions," arXiv:2502.12557, 2025.

[27] B. Sun, M. Theile, Z. Qin, D. Bernardini, D. Roy, A. Bastoni, and M. Caccamo, "Edge generation scheduling for DAG tasks using deep reinforcement learning," arXiv:2308.14647, 2023.

[28] N. Gholipour, M. D. de Assuncao, P. Agarwal, J. Gascon-Samson, and R. Buyya, "TPTO: A Transformer-PPO based task offloading solution for edge computing environments," arXiv:2312.11739, 2023.

[29] P. Phan-Trung *et al.*, "GTPO: A stabilizing task offloading technique based on GTrXL and PPO in IoT edge computing environments," in *Proc. IEEE Conf.*, 2024.

[30] M. Han, X. Sun, X. Wang, W. Zhan, and X. Chen, "Transformer-based distributed task offloading and resource management in cloud-edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 9, pp. 2938–2953, 2025.

[31] Z. Wang, M. Goudarzi, and R. Buyya, "TF-DDRL: A transformer-enhanced distributed DRL technique for scheduling IoT applications in edge and cloud computing environments," *IEEE Trans. Serv. Comput.*, vol. 18, no. 2, pp. 1039–1053, 2025.

[32] G. Rjoub, S. Islam, J. Bentahar, M. A. Almaiah, and R. Alrawashdeh, "Enhancing IoT intelligence: A transformer-based reinforcement learning methodology," arXiv:2404.04205, 2024.

[33] D. Byrne, "Double Deep Q-network (DDQN) implementation," PyTorch Lightning Bolts, 2020. [Online]. Available: https://lightning-bolts.readthedocs.io/en/stable/models/reinforce_learn.html

XraySafe-YOLO: Compact Scale-Aware Fusion and Contour-Motivated Attention for Recall-Oriented Detection of Occluded Prohibited Items in X-ray Images

Liu Liu, Wei Huang*
Department of General Education, Shanghai Customs College, Shanghai, China

***Abstract*—Occlusion, weak texture, and scale variation make prohibited-item detection difficult in X-ray security images. This study presents XraySafe-YOLO, a compact detector built on YOLO11n that combines Contour-Guided Occlusion Attention with Lightweight Scale-Aware Feature Fusion. On OPIXray, at a fixed confidence threshold of 0.25 and intersection-over-union threshold of 0.50, the full model reduced false negatives from 201 to 163 in the seed-42 experiment. Across three split-and-training trials, the full model reduced false negatives and improved fixed recall over YOLO11n in every trial, with a mean false-negative count of 172.7 and a mean fixed recall of 0.9030. A supplemental single-seed evaluation on the 12-class PIDray benchmark increased mean average precision at an intersection-over-union threshold of 0.50 from 0.7938 to 0.8035 and reduced false negatives from 10,148 to 9,318. LSAFF produced the more stable individual improvement, whereas CGOA alone was sensitive to the OPIXray trial seed. Results on both datasets show the same main behavior: the proposed combination improves recall and reduces missed detections, while some false-positive and precision costs remain. The method is therefore positioned as a compact recall-oriented option for human-assisted X-ray security inspection rather than as a detector that improves every metric.**

***Keywords*—X-ray imaging; prohibited item detection; object detection; YOLO; occlusion-aware detection; multi-scale feature fusion; recall-oriented evaluation**

I. Introduction

X-ray imaging is widely used to inspect baggage, parcels, and cargo at airports, ports, railway stations, logistics centers, and other security checkpoints. It allows inspectors to check concealed objects without opening the container. However, detection becomes difficult when prohibited items are small, have weak contrast, or overlap with normal objects. A high inspection workload can also increase the chance of missed detections.

Automatic detection can help inspectors by marking regions that need further checking. X-ray images are different from normal color images. They have limited texture, incomplete object boundaries, and overlapping materials. Downsampling may also weaken small-object features. These conditions make false negatives an important issue.

*Corresponding author.

Mean average precision is useful for comparing detectors, but it does not show how many prohibited items are missed at a selected threshold. In a human-assisted inspection process, a false positive usually leads to another check, while a false negative may allow a dangerous item to pass. Therefore, this study reports standard detection metrics together with true-positive, false-positive, and false-negative counts, fixed-threshold recall, a recall-weighted F_2 score, occlusion-level results, and model complexity.

YOLO detectors are suitable for inspection tasks because they use a compact one-stage structure. Existing X-ray detection methods often use attention, feature-pyramid changes, larger receptive fields, or efficient convolution. This study focuses on two points. First, local shape and boundary information may still be useful when texture is weak or an object is partly hidden. Second, features from different scales should be combined without making the detector too large.

This study asks three questions. Can compact multi-scale fusion reduce missed detections without replacing the YOLO11n backbone? Does local boundary-motivated attention help by itself, or mainly when it is combined with multi-scale features? Are the results similar across repeated split-and-training trials? We first test the modules separately and together on OPIXray. We then use PIDray to check whether the same behavior appears with more object categories and the official Easy, Hard, and Hidden subsets.

XraySafe-YOLO uses two plug-in modules. Contour-Guided Occlusion Attention (CGOA) applies spatial and channel recalibration to the higher-resolution P3 and P4 features. It is motivated by local boundary changes, but it does not use contour labels or an external edge detector. Lightweight Scale-Aware Feature Fusion (LSAFF) aligns P3, P4, and P5 and learns the contribution of each input scale to each output scale. The modules are tested separately and together.

The main contributions are as follows:

- A compact YOLO11n-based detector is proposed for occluded prohibited items in X-ray images. CGOA applies local spatial reweighting motivated by boundary variation, and LSAFF fuses features for each detection scale.
- A controlled OPIXray ablation compares the two

modules separately and together, with CBAM-lite as an attention reference. The four main configurations are tested in three trials with different validation and training seeds.

- A single-seed second-benchmark test on PIDray compares YOLO11n, YOLO11n + LSAFF, and the full model on the official Easy, Hard, and Hidden subsets.
- The evaluation separates class-averaged standard metrics from micro-averaged fixed-count metrics, so the trade-off between missed detections and false alarms can be seen directly.
- The study reports occlusion-level, class-level, complexity, repeated-trial, operating-point, second-benchmark, and qualitative analyses. Model definitions, configuration files, result summaries, and a small execution demo are available from the corresponding author upon reasonable request.

The remainder of the study is organized as follows. Section II reviews related work. Section III describes the architecture and evaluation design. Section IV presents the experimental protocol, Section V reports the results, and Section VI discusses their practical meaning and limitations. Section VII concludes the study.

II. Related Work

A. Benchmarks and Overlap-Aware X-ray Detection

X-ray prohibited-item detection is different from natural-image detection because overlapping materials, weak texture, and incomplete boundaries can mix object and background information. Public datasets have played an important role in this field. SIXray introduced a large-scale dataset with serious class imbalance and overlapping baggage contents [1]. OPIXray focuses on five cutter-like categories and provides three official occlusion-level test subsets [2], [3]. PIDray includes more categories and hidden objects in realistic scenes [4]–[6].

Recent methods also address overlap directly. AO-DETR uses category-specific assignment and dense decoder refinement to reduce feature coupling and unclear boundaries on PIXray and OPIXray [7]. Dual-view detection shows that two scanner views can help solve cases that remain difficult in one view [8]. These methods are important references, but they use different architectures and, for dual-view detection, different input data from the compact single-view setting used in this study.

B. Lightweight YOLO Detectors for Security X-ray Images

YOLO detectors are widely used when deployment speed and implementation simplicity matter [9]–[13]. X-ray variants have explored attention, feature aggregation, dynamic heads, lightweight backbones, loss redesign, pruning, and distillation. ScanGuard-YOLO emphasizes recall under cluttered and occluded conditions [14]. Fang et al. reduce YOLOv8 channels and introduce a simplified BiFPN to balance accuracy and computation [15]. Fine-YOLO redesigns feature aggregation and uses normalized Wasserstein distance to improve small-object localization [16]. Attention-based X-ray detection has also combined multi-scale features with explicit feature recalibration [17].

The design space has continued to expand. A lightweight YOLOv7-tiny model targets portable inspection devices through efficient architectural changes [18]. Guan et al. combine deformable convolution and a dynamic detection head in YOLOv8 [19]. Chen et al. report another improved YOLOv8 framework for contraband detection [20]. Xray-YOLO-Mamba integrates selective state-space blocks and evaluates across CLCXray, OPIXray, and SIXray [21]. More recent work combines depthwise convolution, channel-spatial attention, and supervised contrastive learning on two X-ray datasets [22], while GSA-YOLO uses structured sparsity and adaptive knowledge distillation to emphasize both accuracy and real-time efficiency [23].

These studies show that a compact X-ray detector should not be judged only by parameter count. Recent work often reports module ablations, computation, visual examples, and tests on more than one dataset or more than one training run. This study has a narrower goal. It examines how local contour-motivated attention and one-stage scale-weighted fusion affect missed detections, false alarms, and trial-to-trial variation in a compact YOLO11n detector.

C. Efficient Attention and Multi-Scale Fusion

Efficient detectors often use depthwise separable convolution and compact channel transformations derived from MobileNet-style designs [24], [25]. Attention modules such as the Convolutional Block Attention Module (CBAM) and coordinate attention recalibrate spatial or channel responses with limited overhead [26], [27]. In X-ray images, this can help emphasize small visible fragments, although clutter can also create strong responses that do not belong to a prohibited item.

Multi-scale fusion addresses the complementary need for fine spatial detail and broader context. Feature Pyramid Networks (FPNs) and Path Aggregation Networks (PANets) exchange information between feature levels, EfficientDet uses repeated bidirectional weighted fusion, and adaptively spatial feature fusion (ASFF) learns scale contributions at the spatial level [28]–[31]. LSAFF follows the same broad motivation but uses one normalized weighted fusion for each output scale rather than a repeated bidirectional pyramid. CGOA is likewise deliberately compact and requires neither contour annotations nor an external edge detector.

D. Evaluation Gap and Positioning of this Study

Most X-ray detection papers report mAP, precision, recall, parameters, FLOPs, and selected examples. These measures are useful, but they may not show how many prohibited items are missed at one selected threshold or whether an improvement is stable across training runs. In a human-assisted workflow, it is also important to know whether lower FN is obtained with more FP.

For this reason, the study reports standard metrics together with class-aware fixed-threshold TP, FP, and FN counts, a recall-weighted F_2 score, occlusion-level and class-level results, three OPIXray trials for the four main configurations,

and a PIDray second-benchmark test. Table I summarizes representative recent work. The absolute scores are not compared across rows because the datasets, classes, resolutions, and evaluation settings are different.

III. Methodology

A. Overview of XraySafe-YOLO

XraySafe-YOLO adds two low-overhead modules to YOLO11n. The modules address incomplete local structure and large changes in object scale. The network keeps the standard one-stage detection pipeline and the original detection heads.

An input image is resized and normalized before entering the backbone. The backbone and neck produce three feature levels, P3, P4, and P5. CGOA is applied to P3 and P4, where spatial detail is still sufficient to represent small or partly visible object boundaries; P5 is left unchanged. In the CGOA-only model, the enhanced P3 and P4 features proceed to the detection heads through the original neck. In the combined model, P3′, P4′, and P5 are passed to LSAFF, which returns $P3^{out}$, $P4^{out}$, and $P5^{out}$ for the three heads. The heads predict bounding boxes, categories, and confidence scores.

Both modules are designed as plug-ins rather than as a replacement for the YOLO11n backbone. Fig. 1 summarizes the data flow.

B. Baseline Detector and Model Variants

YOLO11n is used as the compact baseline. All variants retain its backbone, neck, and detection heads, which keeps the comparison focused on the added modules. Five configurations are evaluated at seed 42: YOLO11n, YOLO11n + CBAM-lite, YOLO11n + CGOA, YOLO11n + LSAFF, and YOLO11n + CGOA + LSAFF. The four principal configurations excluding CBAM-lite are also evaluated with seeds 42, 2026, and 3407.

The implementation places CGOA after the final P3 and P4 neck branches; P5 bypasses the module. In the model definition, these features correspond to layers 16, 19, and 22. LSAFF is instantiated once for each target resolution so that the same P3/P4/P5 inputs produce one fused output for each detection head. Shape-compatible YOLO11n parameters are transferred from the pretrained checkpoint to each custom configuration, while parameters introduced by CGOA, LSAFF, and CBAM-lite use their module initializations. Input size, batch size, epoch limit, data preparation, optimizer policy, augmentation, early stopping, and fixed-count evaluation settings are kept the same across models; no architecture-specific hyperparameter tuning is performed.

C. Contour-Guided Occlusion Attention Module

X-ray images usually contain less texture than natural images. However, the shape and boundary of a prohibited item may still be visible when the item is partly covered. For example, part of a knife blade or the outline of scissors can still provide useful information in cluttered baggage. CGOA is based on this observation. In the module name, "guided" means that the design is guided by local boundary variation; it does not use an external contour image or contour supervision.

Let $F \in \mathbb{R}^{C\times H\times W}$ denote an intermediate feature map extracted by the backbone. CGOA is inserted into the P3 and P4 branches because these feature levels preserve more spatial detail and are more relevant to small or partially occluded prohibited items. The module first generates a local spatial response map from F. This response is contour-motivated rather than contour-supervised: no edge labels, contour annotations, or external edge detector are used. Specifically, the spatial branch applies a 3×3 depthwise convolution with batch normalization and SiLU activation, followed by a 1×1 convolution that reduces the response to one channel. The contour branch produces a spatial logit map $M_c \in \mathbb{R}^{1\times H\times W}$.

The contour response is then combined with channel-wise recalibration. Given the feature map F, global average pooling and global max pooling are used to obtain channel descriptors. The two descriptors pass through the same two-layer 1×1 convolutional MLP with SiLU activation. Its hidden width is $\max(\lfloor C/16 \rfloor, 8)$. The summed MLP output is denoted by L_{ch}, and the channel weights are $A_{ch} = \sigma(L_{ch}) \in [0,1]^{C\times 1\times 1}$.

The enhanced feature map is computed as follows:

$$F' = F \otimes (1 + \sigma(M_c)) \otimes A_{ch}, \tag{1}$$

where, $\otimes$ denotes element-wise multiplication with broadcasting and $\sigma(\cdot)$ denotes the sigmoid activation function. The residual spatial factor $1 + \sigma(M_c)$ is never smaller than one, so the spatial branch alone does not attenuate F; however, the subsequent channel gate $A_{ch} \in [0,1]$ can attenuate channels in the final product. CGOA is restricted to P3 and P4 because these feature maps retain finer boundary information; applying the same spatial gate to the coarser P5 branch would add computation where local contour detail is already strongly downsampled.

The residual spatial gate is intended to emphasize local transitions without directly suppressing them; the complete CGOA output can still be attenuated by the channel gate. Thus, boundary enhancement is treated as a design motivation rather than as a demonstrated contour detector. Because X-ray clutter can also create strong local transitions, the ablation study evaluates whether this bias helps consistently rather than assuming that it will do so. Fig. 2 illustrates the CGOA module.

D. Lightweight Scale-Aware Feature Fusion Module

Prohibited items can have different sizes and irregular shapes. Small items need high-resolution details, while large or heavily occluded items need more context. Multi-scale feature fusion is therefore useful. However, some fusion structures add many parameters and computations, which is not suitable for a compact baseline.

LSAFF addresses this trade-off with one output-specific weighted fusion stage for each detection scale. For every input-to-output path, the implementation uses the same architectural pattern: a 3×3 depthwise convolution followed by a 1×1 pointwise convolution, with batch normalization and SiLU activation after each convolution. The resulting feature is resized to the target spatial resolution with nearest-neighbor interpolation. The three output-specific LSAFF instances have

TABLE I. REPRESENTATIVE RECENT DIRECTIONS IN X-RAY PROHIBITED-ITEM DETECTION AND THE POSITION OF THIS STUDY

Method	Year	Evaluation Data	Main Technical Emphasis	Evidence Emphasized
AO-DETR [7]	2025	PIXray, OPIXray	Category-specific assignment and dense decoder refinement for overlap	Multi-dataset accuracy, overlap analysis, qualitative localization
Lightweight YOLOv8 [15]	2024	PIDray, CLCXray	Channel reduction, simplified BiFPN, and lightweight loss design	Accuracy–complexity balance and second-benchmark validation
Fine-YOLO [16]	2024	EDS, NEU-DET	Lightweight feature aggregation and normalized Wasserstein distance	Small-object accuracy, complexity, auxiliary validation
Xray-YOLO-Mamba [21]	2025	CLCXray, OPIXray, SIXray	State-space feature extraction and lightweight sampling blocks	Three-dataset results, FLOPs, parameters, FPS, heatmaps
YOLOv8-DWConv-CSAF [22]	2026	CLCXray, HiXray	Depthwise convolution, channel-spatial attention, contrastive learning	Two-dataset validation, efficiency, multi-scale analysis
GSA-YOLO [23]	2026	HiXray, PIDray	Structured sparsity and adaptive knowledge distillation	Cross-dataset accuracy and real-time inference
XraySafe-YOLO	–	OPIXray, PIDray	CGOA, LSAFF, and controlled YOLO11n ablations	Fixed TP/FP/FN, three-trial variability, occlusion, class, and second-benchmark analysis

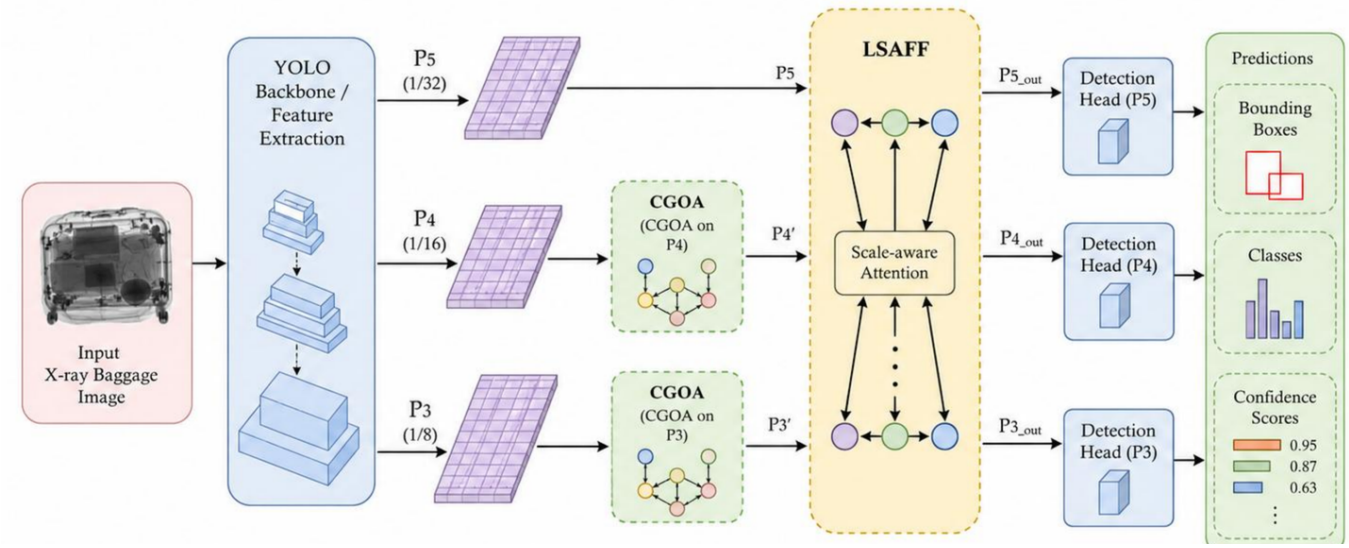


Fig. 1. Overall architecture of the proposed XraySafe-YOLO framework. The YOLO backbone extracts three feature maps, P3, P4, and P5. CGOA is applied only to the P3 and P4 branches to reweight local spatial responses, motivated by boundary transitions in small and partially occluded prohibited items, while P5 bypasses CGOA and enters LSAFF directly. LSAFF then produces learned scale-aware output features $P3^{out}$, $P4^{out}$, and $P5^{out}$ for the three detection heads, which output bounding boxes, classes, and confidence scores.

independent parameters; the transformations are not shared across target scales.

Let F_i denote the feature map from the i-th input scale, where $i \in \{3,4,5\}$. Unlike a single-output fusion block, LSAFF produces three scale-specific output features for the detection heads. The transformation from input scale i to output scale k is denoted by $\phi_{k,i}(\cdot)$. For each output scale $k \in \{3,4,5\}$, LSAFF assigns normalized learned scale weights $\alpha_{k,i}$ to the aligned input features. Let the weighted intermediate fusion be:

$$\widetilde{F}_k = \sum_{i\in\{3,4,5\}} \alpha_{k,i} \cdot \phi_{k,i}(F_i), \quad k \in \{3,4,5\}, \tag{2}$$

and let $\psi_k(\cdot)$ denote the output-specific depthwise-separable refinement block. The feature delivered to the k-th detection head is

$$F_k^{out} = \psi_k\left(\widetilde{F}_k\right). \tag{3}$$

Here, $\alpha_{k,i}$ is a learned scalar contribution of the i-th input scale to the k-th output scale. Each output-specific LSAFF instance owns an independent three-element logit vector initialized to zero, corresponding to equal initial contributions from P3, P4, and P5. The weights are normalized for each output scale:

$$\alpha_{k,i} = \frac{\exp(w_{k,i})}{\sum_{j\in\{3,4,5\}} \exp(w_{k,j})}. \tag{4}$$

The refinement ψ_k is a second 3×3 depthwise-separable convolution applied after weighted summation. The weights are learned independently for each output scale; they are scalar coefficients rather than image-dependent or spatially

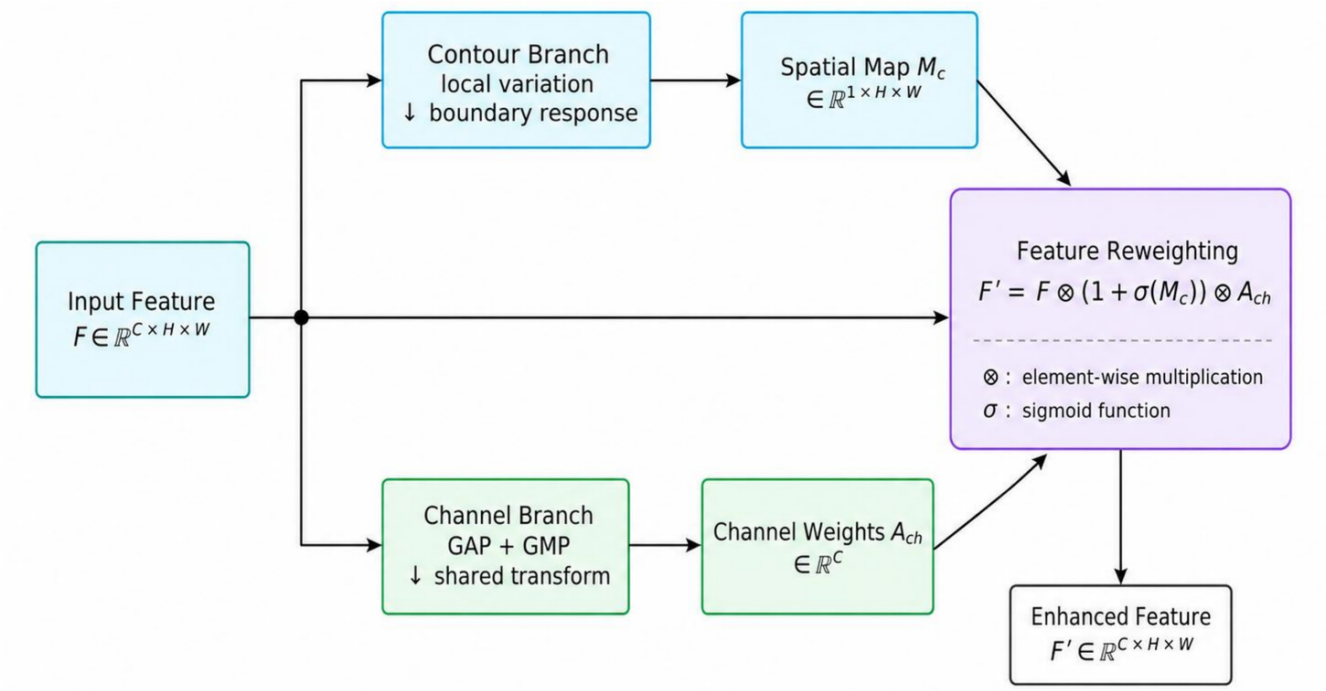


Fig. 2. Contour-Guided Occlusion Attention (CGOA) module. Given an input feature map F, the contour-motivated spatial branch generates a response map M_c, the channel branch generates channel weights A_{ch} using global average and max pooling, and the identity path preserves the original feature. The enhanced output is computed as $F' = F \otimes (1 + \sigma(M_c)) \otimes A_{ch}$ without requiring additional contour annotations.

varying attention maps. High-resolution features retain small-object and boundary information, while deeper features provide broader context. Unlike bidirectional pyramids such as BiFPN, which repeatedly propagate features in top-down and bottom-up paths, LSAFF performs one weighted fusion for each output scale. This keeps the added computation modest, although no direct LSAFF–BiFPN performance comparison is made in this study. Fig. 3 illustrates the module.

E. Training Objective and Recall-Oriented Evaluation

The model uses the standard YOLO detection loss. This keeps the comparison focused on the proposed modules and avoids adding another change through the loss function. The training loss follows the common YOLO-style structure:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{dfl}\mathcal{L}_{dfl}, \tag{5}$$

where, $\mathcal{L}_{box}$ is the bounding-box regression loss, $\mathcal{L}_{cls}$ is the classification loss, and $\mathcal{L}_{dfl}$ is the distribution focal loss used by the underlying YOLO training framework [32]. No class-balanced or hard-sample loss is added. This design keeps the ablation focused on the two proposed architectural modules, CGOA and LSAFF.

Because prohibited item detection is sensitive to missed detections, recall orientation is reflected in the evaluation protocol rather than in a separate training loss. In addition to mAP, the analysis reports recall, false-negative count, fixed-threshold recall, occlusion-level recall, and efficiency metrics. These metrics make it possible to determine whether the proposed modules improve difficult X-ray inspection cases without overstating the model as an autonomous safety decision system.

IV. Experimental Protocol

A. Datasets

The primary experiments use the public OPIXray benchmark, which was constructed for occluded prohibited item detection in X-ray security inspection images [2], [3]. OPIXray contains 8885 images from five cutter-like prohibited item categories: folding knife, straight knife, scissors (official label: "scissor"), utility knife, and multi-tool knife. The official test set is divided into three occlusion-level subsets, denoted as OL1, OL2, and OL3, which support a direct analysis of detection behavior under increasing overlap and visual clutter. Table II summarizes the OPIXray split used for the controlled ablation and repeated trials.

A supplemental evaluation uses the current 124,486-image PIDray release distributed through the official `lutao2021/PIDray` repository [4]–[6]. This release contains 76,913 training images and official Easy, Hard, and Hidden test subsets with 24,758, 9,746, and 13,069 images, respectively, across 12 prohibited-item categories. A deterministic 5% holdout from the official training split is used for validation, while all images in the three official test subsets are retained, including images without prohibited-item boxes; predictions on such images therefore contribute false positives when applicable. The converted test annotations contain 8450, 14,500, and 19,050 ground-truth boxes, respectively, for a total of 42,000. Because the models are trained on PIDray training data, this experiment evaluates the architecture on a

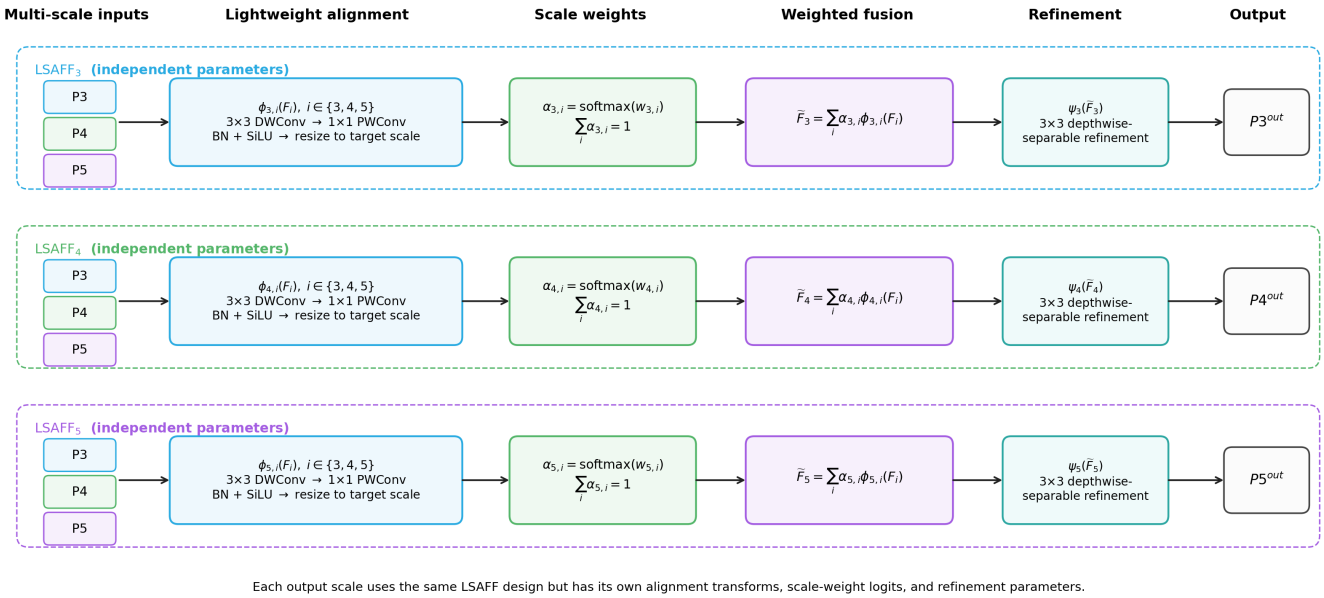


Fig. 3. Lightweight Scale-Aware Feature Fusion (LSAFF) module. Multi-scale inputs P3, P4, and P5 are aligned and transformed through lightweight operations. Three independently parameterized output-specific blocks align P3/P4/P5, learn normalized scale weights $\alpha_{k,i}$, form $\widetilde{F}_k$, and apply the refinement ψ_k before producing P3^{out}, P4^{out}, and P5^{out}.

second benchmark rather than cross-dataset generalization of OPIXray-trained weights.

Representative OPIXray examples are shown in Fig. 4. The examples illustrate why prohibited item detection in X-ray images is more difficult than ordinary natural-image detection: item contours can be weak, objects can be transparent or overlapping, and cluttered baggage contents may visually interfere with cutter-like objects.

B. Data Preparation

OPIXray is converted into a unified YOLO-compatible annotation format. Each image is associated with a text annotation file containing the object category and normalized bounding-box coordinates. The official OPIXray test set is preserved in every trial. For the seed-42 architecture comparison, 711 images are selected from the official training portion using split seed 42. For the repeated seeded evaluation, the 711-image validation subset is regenerated using split seeds 42, 2026, and 3407, respectively; the official test set remains unchanged.

The conversion process includes parsing original annotation files, mapping dataset-specific categories to unified class names, converting bounding-box coordinates to normalized YOLO format, verifying image-annotation consistency, generating train/validation/test file lists, and checking invalid boxes, missing images, and duplicated annotations.

PIDray is converted from its COCO-style detection annotations to the same YOLO-compatible format while retaining all 12 official categories. The training images are partitioned deterministically into training and validation sets with a 5% validation ratio and seed 42. Easy, Hard, and Hidden remain separate test sets. The combined PIDray standard metrics are obtained by running the validator once on the union of all three official test subsets (pooled predictions), not by averaging the three subset mAP values. Fixed TP/FP/FN counts are summed over the same pooled test population.

C. Implementation Details

The experiments were run on Windows with Python 3.11, PyTorch, Ultralytics YOLO 8.4.78, and an NVIDIA GeForce RTX 4070 SUPER GPU with 12 GB graphics memory. Images were resized to 640×640, the batch size was 48, and training was allowed to continue for up to 100 epochs with early-stopping patience 30. All models used pretrained YOLO11n initialization, cosine learning-rate scheduling, weight decay 0.0005, three warm-up epochs, deterministic seed settings, and mixed-precision training. Standard Ultralytics augmentations were retained, including HSV adjustment, translation, scaling, horizontal flipping, and mosaic augmentation with `close_mosaic=10`. No separate RandAugment or random-erasing pipeline was added.

The same Ultralytics training interface and `optimizer=auto` policy were used for every model, and no model received architecture-specific optimizer tuning. Under Ultralytics 8.4.78, the OPIXray runs resolved to AdamW with initial learning rate 0.001111 and $\beta_1 = 0.9$; cosine scheduling used the framework default final learning-rate factor 0.01. Weight decay was 0.0005 and warm-up lasted three epochs. The larger PIDray runs resolved to MuSGD with initial learning rate 0.01 and momentum 0.9 under the same automatic-selection rule. Shape-compatible pretrained parameters and newly initialized module parameters were optimized together. These resolved settings are reported to make clear that `optimizer=auto` was not an unspecified degree of freedom.

The seed-42 OPIXray comparison used split seed 42 and training seed 42 for all five models. For the repeated OPIXray

TABLE II. OPIXRAY DATASET STATISTICS USED FOR THE PRIMARY EXPERIMENTS

Dataset	Images	Classes	Train	Val	Test	Test GT Boxes by OL Subset
OPIXray	8885	5	6398	711	1776	OL1: 923; OL2: 550; OL3: 307

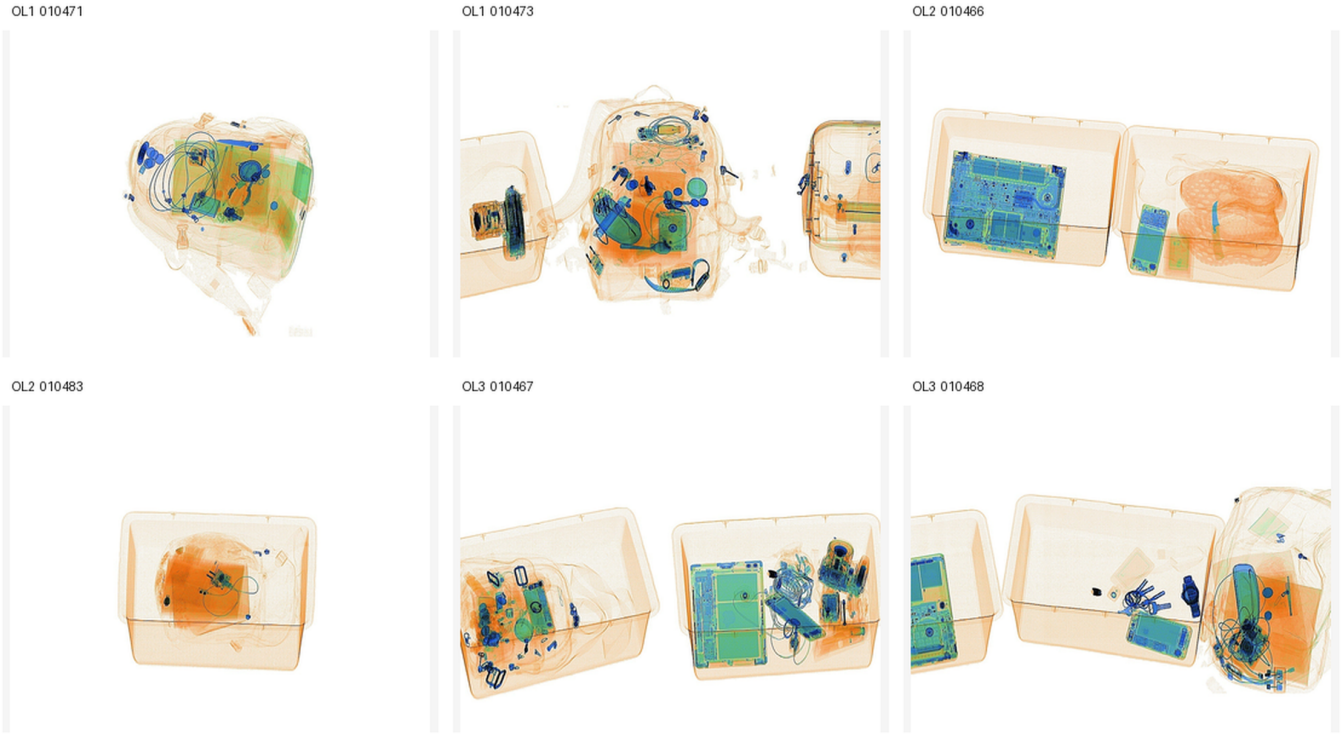


Fig. 4. Representative OPIXray examples from OL1, OL2, and OL3 occlusion-level subsets. The images show the transparent appearance, weak texture, and object overlap that motivate contour-motivated and recall-oriented detection analysis.

evaluation, the four principal models were trained with seeds 42, 2026, and 3407. In each trial, the validation split seed matched the training seed, while the official OPIXray test set remained unchanged. The repeated results therefore combine split and optimization variability; they are not a fixed-split significance test. Hardware-dependent validation timing was recorded only as a diagnostic and is not used to support the architectural conclusions.

The supplemental PIDray experiment used the same software and hardware environment, 640-pixel input resolution, pretrained initialization, augmentation family, optimizer policy, early-stopping mechanism, and fixed-count thresholds. Because of the larger dataset, training used batch size 16 for up to 50 epochs with split and training seed 42. YOLO11n, YOLO11n + LSAFF, and the full model were trained under the same PIDray protocol. The experiment uses one seed and is interpreted as second-benchmark evidence rather than a significance test.

For the seed-42 OPIXray architecture comparison, training completed 100 epochs for YOLO11n and CGOA, 94 epochs for LSAFF, and 92 epochs for the full model because early stopping was triggered for the latter two runs. The reported test metrics use the saved `best.pt` checkpoint selected by the Ultralytics validation fitness, which weights mAP@0.5 and mAP@0.5:0.95 by 0.1 and 0.9, respectively. During fixed-count inference, the prediction confidence threshold is varied as stated in each analysis; non-maximum suppression uses IoU 0.50, class-aware suppression (`agnostic_nms=False`), and `max_det=300`. The subsequent TP/FP/FN matcher separately uses same-class IoU $\geq$ 0.50. Thus, the NMS IoU threshold and the matching IoU threshold are numerically equal in this study but serve different operations (Table III).

TABLE III. RESOLVED TRAINING SETTINGS FOR THE PRINCIPAL CONFIGURATIONS

Setting	YOLO11n	CGOA	LSAFF	Full
Optimizer (OPIXray)	AdamW	AdamW	AdamW	AdamW
Initial LR	0.001111	0.001111	0.001111	0.001111
β_1 / momentum	0.9	0.9	0.9	0.9
Weight decay	0.0005	0.0005	0.0005	0.0005
Completed epochs (seed 42)	100	100	94	92
Checkpoint	best validation fitness: 0.1 mAP@0.5 + 0.9 mAP@0.5:0.95			

For PIDray, the same `optimizer=auto` rule resolves to MuSGD because of the larger number of planned iterations; its initial learning rate is 0.01 with momentum 0.9. The PIDray

runs use the same weight decay, cosine scheduling, warm-up policy, checkpoint rule, and inference settings unless otherwise stated.

D. Evaluation Metrics

Standard object detection metrics are reported:

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{8}$$

In addition, mAP@0.5 and mAP@0.5:0.95 are reported. The standard precision, recall, and F1 values are class-averaged metrics returned by the Ultralytics validator. They are therefore not numerically interchangeable with the fixed-count metrics below, which are micro-averaged over all 1780 test instances at one selected confidence and IoU operating point. Per-class recall and AP values are retained in the metric JSON files and summarized for the baseline and full model to inspect whether the aggregate gain is concentrated in only one category.

Because missed prohibited items are important in security inspection, additional recall-oriented metrics are also reported. These include false-negative count, occlusion-level recall, and fixed-threshold recall. Deployment-oriented complexity metrics include parameter count, GFLOPs, and model size. These metrics describe architectural overhead while avoiding unsupported claims about hardware-dependent latency or throughput.

The fixed-count protocol is computed after the model's standard prediction and non-maximum suppression stage. Predictions are filtered with confidence ≥ 0.25 and sorted by descending confidence. Each prediction is greedily matched to the unmatched ground-truth box of the same class with the highest IoU, provided that IoU ≥ 0.50. Each ground-truth instance can be matched at most once. Unmatched predictions are counted as false positives, and unmatched ground-truth instances are counted as false negatives. Fixed-count precision and recall are then computed as:

$$Fixed\ Precision = \frac{TP_{fixed}}{TP_{fixed} + FP_{fixed}}, \tag{9}$$

$$Fixed\ Recall = \frac{TP_{fixed}}{TP_{fixed} + FN_{fixed}}. \tag{10}$$

Because missed detections are emphasized in the intended human-in-the-loop screening setting, a recall-weighted fixed-count F_2 score is also reported; with $\beta = 2$, recall receives four times the weight of precision in the denominator-weighted harmonic formulation:

$$F_2 = \frac{5 \cdot Fixed\ Precision \cdot Fixed\ Recall}{4 \cdot Fixed\ Precision + Fixed\ Recall}. \tag{11}$$

For the seed-42 OPIXray test set, this protocol counts 1776 test images and 1780 ground-truth boxes. For PIDray, the same protocol is applied to the 42,000 ground-truth boxes across the official Easy, Hard, and Hidden subsets. The 0.25 confidence threshold is a pre-specified diagnostic point applied identically to every model rather than a claim of universal optimality. Operational deployment would require threshold calibration on data from the target scanner and analysis of the full precision–recall curve.

E. Baseline and Comparative Models

The main OPIXray ablation compares YOLO11n, YOLO11n + CGOA, YOLO11n + LSAFF, and YOLO11n + CGOA + LSAFF. This arrangement isolates the two modules within the same detector and training protocol. CBAM-lite is included at seed 42 as a generic channel-and-spatial attention reference trained under the same settings. The compact PIDray comparison includes YOLO11n, YOLO11n + LSAFF, and the full model. CGOA alone is not repeated on PIDray because the OPIXray trials already identify LSAFF as the stronger standalone component; the second-benchmark experiment focuses on whether the same architectural pattern is observed on a broader 12-class benchmark.

F. Visualization and Descriptive Analysis

The numerical tables are supported by figures generated directly from the reported test results. Error bars in the repeated-trial figures denote the sample standard deviation over seeds 42, 2026, and 3407. The OPIXray occlusion figure uses the fixed FN counts from the official OL1–OL3 subsets, and the class figure plots the recall difference between the full model and YOLO11n. The PIDray second-benchmark figure compares mAP@0.5 and fixed recall across Easy, Hard, and Hidden. These visual summaries do not introduce additional samples or alternative metric definitions; they are intended to make variability, trade-offs, category-specific behavior, and second-benchmark behavior easier to inspect.

G. Reproducibility and Consistency Checks

Dataset split seeds, training seeds, model definitions, conversion summaries, and result summaries were retained for each experiment. A compact code package available from the corresponding author contains the CGOA, LSAFF, and CBAM-lite modules, the associated YAML files, and a CPU-compatible tensor-shape smoke test. The original OPIXray and PIDray images and annotations are not redistributed and must be obtained from their official sources under the applicable academic-use terms.

The fixed-count tables were checked directly from their TP, FP, FN, and ground-truth totals. Every row satisfies $TP + FN = GT$, and fixed precision and recall were recomputed from the counts. For the occlusion analysis, the OL1–OL3 TP and FN values sum to the full 1780-instance test set for each model. These checks place the aggregate and subset TP/FN results on the same ground-truth basis.

V. Results and Analysis

A. Controlled Seed-42 Architecture Comparison

Table IV reports the conventional seed-42 metrics. The ranking depends on the metric: CBAM-lite has the highest precision, while LSAFF has the highest recall, F1-score, mAP@0.5, and mAP@0.5:0.95. The combined model is close to LSAFF but does not lead any conventional metric. Its class-averaged recall (0.8439) is the arithmetic mean of the five class recalls in Table VII; F1 is recomputed from the corresponding precision and recall. CGOA alone performs below YOLO11n in this trial. Among the proposed individual modules, LSAFF provides the stronger seed-42 result.

The fixed operating-point counts in Table V show the trade-off between missed detections and false alarms more clearly than mAP alone. Relative to YOLO11n, the seed-42 full model detects 38 additional ground-truth instances (1617 versus 1579), reducing FN from 201 to 163, or by 18.9%. This recall gain is accompanied by an increase in FP from 224 to 250, or by 11.6%. At this operating point, each added false positive is associated with about 1.46 additional true detections. Fixed recall rises from 0.8871 to 0.9084, fixed precision decreases from 0.8758 to 0.8661, and the recall-weighted F_2 increases from 0.8848 to 0.8996. The ratio is not a deployment utility measure because the costs of misses and false alarms are application-dependent; it simply makes the observed count trade-off easier to interpret. Therefore, the combined model is better described as a recall-oriented option, not as a model that improves every error type or metric.

At seed 42, adding CGOA to LSAFF reduces FN from 184 to 163 and increases fixed recall from 0.8966 to 0.9084, but FP changes from 255 to 250 and fixed precision remains below the baseline. CGOA alone is worse than YOLO11n in FN and conventional mAP for this seed. The result suggests that CGOA can complement LSAFF at this operating point, but it does not support CGOA as an independently superior attention block.

B. Occlusion-Level and Per-Class Behavior

Table VI reports the seed-42 results on OL1, OL2, and OL3. The three subsets contain 923, 550, and 307 ground-truth instances, respectively, and their TP and FN counts sum exactly to the 1780-instance full test set for every model. The full model gives the lowest FN on OL1 and OL3, whereas LSAFF gives the lowest FN on OL2. The results suggest that multi-scale fusion is the more stable contributor and that the combined configuration is most useful when missed detections receive greater weight.

Fig. 5 visualizes the fixed FN counts. Because OL1, OL2, and OL3 contain different numbers of ground-truth boxes, bar heights should be compared between models within a subset rather than across subsets. Relative to YOLO11n, the full model reduces FN by 20 on OL1, 7 on OL2, and 11 on OL3. In fixed-recall terms, these changes correspond to gains of 0.0217, 0.0127, and 0.0358, respectively. LSAFF records the lowest FN on OL2, while the full model is lowest on OL1 and OL3. The largest proportional recall gain occurs on OL3, although this subset is also the smallest and should not be treated as an independent generalization test.

Table VII reports a compact class-level recall comparison between YOLO11n and the full configuration. The full model improves recall for multi-tool knife, the official "scissor" label, straight knife, and utility knife, but decreases recall for folding knife. The gain is especially visible for multi-tool knife and utility knife, whereas straight knife remains the most difficult category. The table focuses on recall because the class-level analysis concerns missed detections. Supporting class-level outputs are available from the corresponding author upon reasonable request.

Fig. 6 shows that the improvement is different across classes. Multi-tool knife has the largest recall increase (+0.0408), followed by utility knife (+0.0196) and straight knife (+0.0165). The small gain for scissors (+0.0039) occurs on a class that already has high baseline recall. Folding knife decreases by 0.0118, showing that the aggregate gain is not uniform; its compact folded geometry may be confused with dense metallic clutter, although the present results do not isolate a single causal factor. Straight knife remains the most difficult class even after improvement, with recall increasing only from 0.5660 to 0.5825. Its elongated, low-contrast appearance and frequent overlap provide a plausible explanation for the persistent difficulty, but this interpretation should be treated as qualitative rather than causal.

C. Repeated Split-and-Training Evaluation

Table VIII summarizes the repeated split-and-training evidence for all four principal models, and Table IX reports the individual trial values. Each trial uses the same batch size, image size, epoch limit, and evaluation protocol, while the validation split seed equals the training seed. The results are therefore descriptive repeated-holdout evidence rather than a fixed-split significance test.

Table IX reports each trial together with the paired change relative to YOLO11n under the same split-and-training seed. This pairing is descriptive rather than a fixed-split significance test, but it shows directly that LSAFF and the full model reduce FN and raise fixed recall in all three trials.

The three trials show different tendencies for the two modules. CGOA has the largest variation in mAP and FN and does not improve the mean values relative to YOLO11n. LSAFF improves mAP@0.5 and fixed recall and lowers FN in each of the three trials, making it the more consistent standalone module. The combined model also lowers FN and raises fixed recall relative to YOLO11n in every trial, while fixed precision is slightly lower in every trial. Averaged over the three trials, its FN count is 15.9% below the baseline mean (172.7 versus 205.3), fixed recall is higher by 0.0184, and fixed precision is lower by 0.0054. The gain is therefore repeatable in direction but concentrated on recall rather than on all operating-point metrics.

Fig. 7 shows the same pattern in count form. The mean FN decreases by 19.3 for LSAFF and by 32.6 for the full model relative to YOLO11n, whereas CGOA changes the mean by only +0.7. CGOA also has the largest FN standard deviation (31.8), consistent with the poor seed-42 result and stronger results in the other two trials. The error bars overlap, so the figure is descriptive evidence rather than a claim of statistical significance.

TABLE IV. SEED-42 BATCH-48 CONVENTIONAL DETECTION METRICS ON THE OPIXRAY TEST SET. BOLD VALUES INDICATE THE BEST RESULT IN EACH COLUMN

Model	Precision	Recall	F1	mAP@0.5	mAP@0.5:0.95
YOLO11n	0.8744	0.8301	0.8517	0.8852	0.4123
YOLO11n + CBAM-lite	**0.8945**	0.8372	0.8649	0.8904	0.4146
YOLO11n + CGOA	0.8512	0.7977	0.8232	0.8467	0.3818
YOLO11n + LSAFF	0.8725	**0.8588**	**0.8650**	**0.8927**	**0.4151**
YOLO11n + CGOA + LSAFF	0.8766	0.8439	0.8599	0.8873	0.4140

TABLE V. SEED-42 FIXED OPERATING-POINT RESULTS AT CONFIDENCE 0.25 AND IOU 0.50. TP, FP, AND FN ARE OBTAINED BY CLASS-AWARE GREEDY MATCHING AFTER NON-MAXIMUM SUPPRESSION

Model	TP	FP	FN	Fixed P	Fixed R	F_2
YOLO11n	1579	224	201	0.8758	0.8871	0.8848
YOLO11n + CBAM-lite	1572	227	208	0.8738	0.8831	0.8812
YOLO11n + CGOA	1540	238	240	0.8661	0.8652	0.8654
YOLO11n + LSAFF	1596	255	184	0.8622	0.8966	0.8895
YOLO11n + CGOA + LSAFF	**1617**	250	**163**	0.8661	**0.9084**	**0.8996**

TABLE VI. OCCLUSION-LEVEL ANALYSIS ON THE OFFICIAL OPIXRAY TEST SUBSETS. RECALL IS THE STANDARD DETECTOR RECALL COMPUTED ON EACH SUBSET; FN IS COUNTED USING THE FIXED CONFIDENCE/IOU PROTOCOL

Model	OL1 Recall / FN	OL2 Recall / FN	OL3 Recall / FN	OL1 mAP@0.5	OL2 mAP@0.5	OL3 mAP@0.5
YOLO11n	0.8574 / 86	0.8081 / 67	0.7787 / 48	**0.9052**	0.8726	0.8609
YOLO11n + CGOA	0.8132 / 104	0.7927 / 75	0.7501 / 61	0.8656	0.8408	0.8084
YOLO11n + LSAFF	0.8634 / 87	**0.8404** / **58**	**0.8398** / 39	0.8898	**0.8957**	**0.8840**
YOLO11n + CGOA + LSAFF	**0.8734** / **66**	0.8342 / 60	0.8150 / **37**	0.9027	0.8737	0.8774

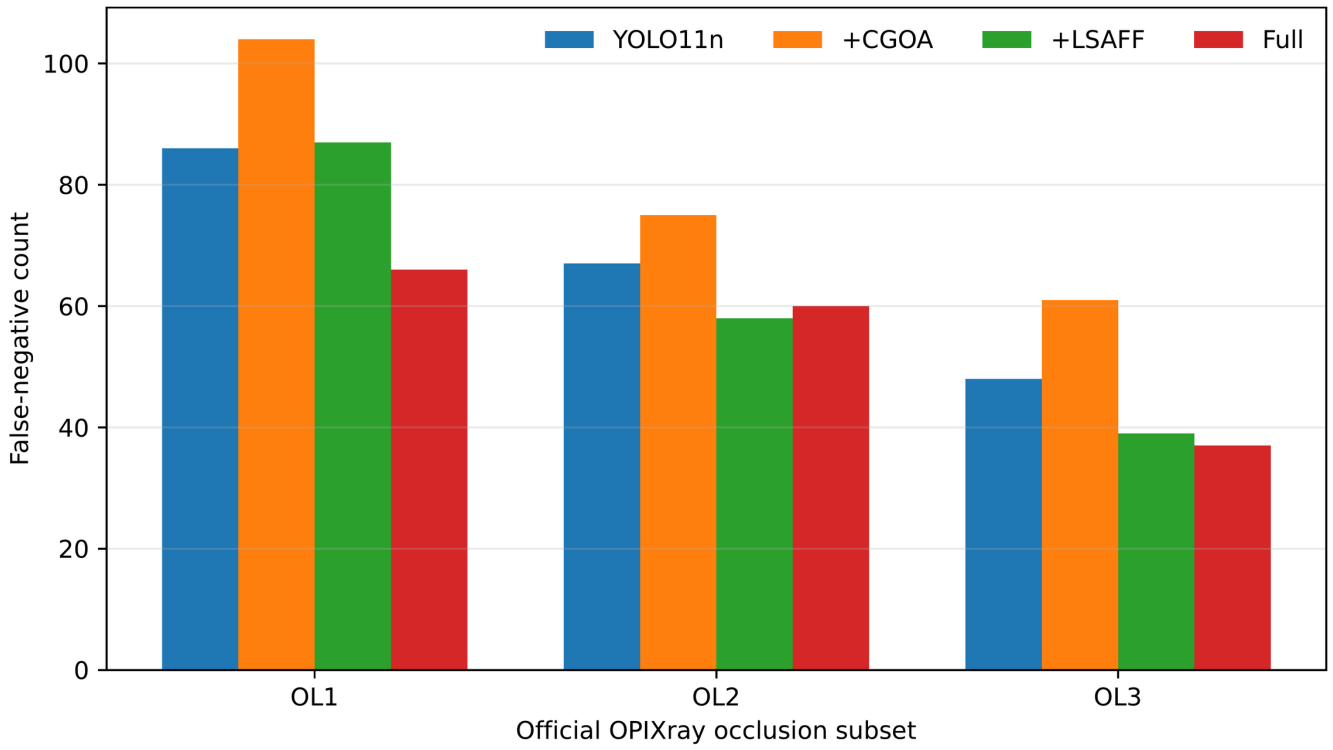


Fig. 5. Fixed false-negative counts on the official OPIXray occlusion subsets at seed 42. The subsets contain different numbers of ground-truth boxes, so comparisons are made between models within each subset.

TABLE VII. CLASS-LEVEL RECALL COMPARISON BETWEEN YOLO11N AND YOLO11N + CGOA + LSAFF ON OPIXRAY

Class	YOLO11n Recall	Full Recall	Δ Recall
folding knife	0.8985	0.8867	-0.0118
multi-tool knife	0.8795	0.9203	0.0408
scissor	0.9755	0.9794	0.0039
straight knife	0.5660	0.5825	0.0165
utility knife	0.8309	0.8505	0.0196

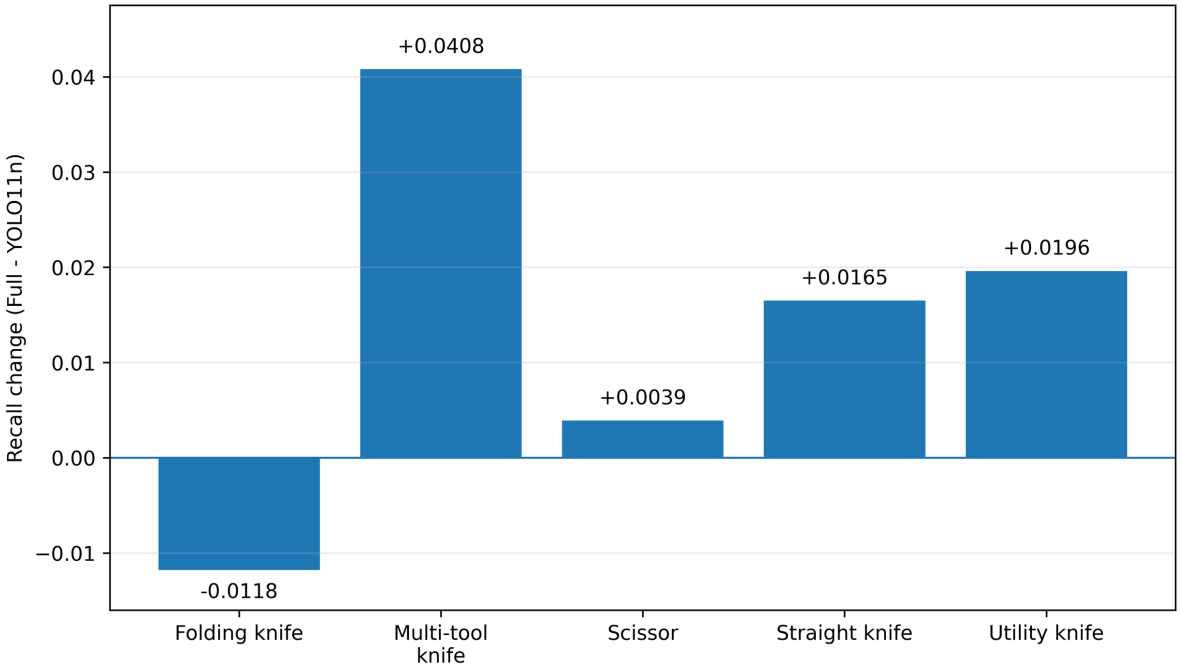


Fig. 6. Class-level recall change of the full model relative to YOLO11n at seed 42. Positive values indicate higher recall for the full model; the folding-knife class shows a small decrease.

TABLE VIII. REPEATED SPLIT-AND-TRAINING SUMMARY OVER SEEDS 42, 2026, AND 3407. VALUES ARE MEAN ± SAMPLE STANDARD DEVIATION; LOWER FN IS BETTER

Model	mAP@0.5	FN	Fixed Recall	Fixed Precision
YOLO11n	0.8814 ± 0.0100	205.3 ± 15.9	0.8846 ± 0.0090	0.8716 ± 0.0214
YOLO11n + CGOA	0.8738 ± 0.0234	206.0 ± 31.8	0.8843 ± 0.0179	**0.8818 ± 0.0135**
YOLO11n + LSAFF	0.8891 ± 0.0102	186.0 ± 15.1	0.8955 ± 0.0085	0.8681 ± 0.0214
YOLO11n + CGOA + LSAFF	**0.8899 ± 0.0114**	**172.7 ± 19.4**	**0.9030 ± 0.0109**	0.8662 ± 0.0209

Fig. 8 places the mean fixed operating points in precision–recall space. The full model occupies the highest-recall position, while CGOA has the highest mean fixed precision but essentially the same mean fixed recall as YOLO11n. LSAFF lies between these outcomes. This view clarifies why the full model is described as recall-oriented rather than uniformly superior.

D. Supplemental Second-Benchmark Evaluation on PIDray

Table X reports the combined Easy, Hard, and Hidden results for the single-seed PIDray experiment. The full model improves standard precision from 0.7832 to 0.7885, standard recall from 0.7498 to 0.7674, mAP@0.5 from 0.7938 to 0.8035, and mAP@0.5:0.95 from 0.6582 to 0.6698. At the fixed operating point, it detects 830 additional ground-truth instances and reduces FN from 10,148 to 9,318, an 8.18% reduction. The gain is accompanied by 409 additional FP, a 4.36% increase, and a small decrease in fixed precision from 0.7726 to 0.7696. The corresponding descriptive yield is 2.03 additional true detections per additional false positive. Fixed recall increases from 0.7584 to 0.7781 and fixed F_2 from 0.7612 to 0.7764. As with OPIXray, this ratio summarizes the observed count change rather than defining a universal screening utility. The result reproduces the recall-oriented

TABLE IX. INDIVIDUAL RESULTS FOR THE THREE SPLIT-AND-TRAINING TRIALS. FIXED R AND FIXED P USE CONFIDENCE 0.25 AND IOU 0.50

Model	Seed	mAP@0.5	FN	ΔFN	Fixed R	ΔR	Fixed P
YOLO11n	42	0.8852	201	–	0.8871	–	0.8758
YOLO11n	2026	0.8701	223	–	0.8747	–	0.8485
YOLO11n	3407	0.8889	192	–	0.8921	–	0.8906
YOLO11n + CGOA	42	0.8467	240	+39	0.8652	-0.0219	0.8661
YOLO11n + CGOA	2026	0.8879	201	-22	0.8871	+0.0124	0.8896
YOLO11n + CGOA	3407	0.8867	177	-15	0.9006	+0.0085	0.8896
YOLO11n + LSAFF	42	0.8927	184	-17	0.8966	+0.0095	0.8622
YOLO11n + LSAFF	2026	0.8776	202	-21	0.8865	+0.0118	0.8502
YOLO11n + LSAFF	3407	0.8971	172	-20	0.9034	+0.0113	0.8918
YOLO11n + CGOA + LSAFF	42	0.8873	163	-38	0.9084	+0.0213	0.8661
YOLO11n + CGOA + LSAFF	2026	0.8801	195	-28	0.8904	+0.0157	0.8453
YOLO11n + CGOA + LSAFF	3407	0.9023	160	-32	0.9101	+0.0180	0.8872

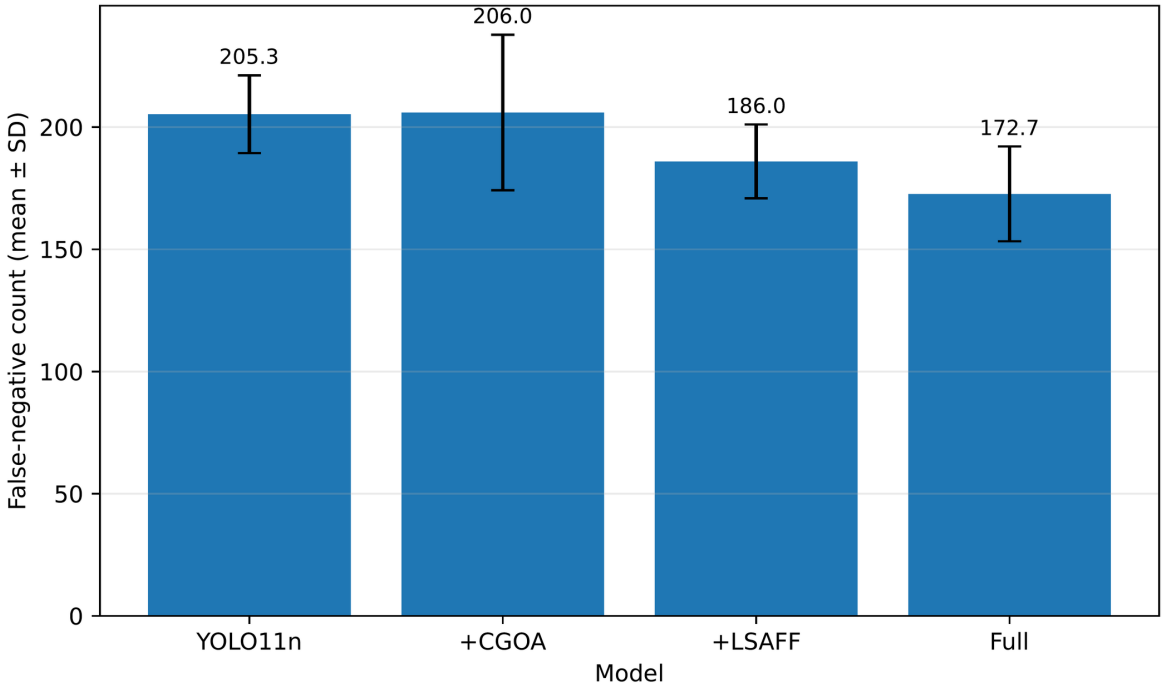


Fig. 7. Mean false-negative count and sample standard deviation over three split-and-training trials, in which both the validation split and training seed change. Lower values are better. The figure summarizes variability and is not a fixed-split significance test.

pattern observed on OPIXray rather than showing a uniform improvement in every error measure.

LSAFF alone already improves the combined PIDray result while keeping fixed precision nearly unchanged (0.7731 versus 0.7726). Adding CGOA within the full model reduces FN by a further 314 relative to LSAFF and gives the best standard recall, mAP, fixed recall, and fixed F_2, although FP increases and fixed precision decreases. This result is consistent with the OPIXray interpretation that LSAFF is the stronger standalone contributor and CGOA is most useful when coupled with multi-scale fusion.

Fig. 9 separates the three official PIDray difficulty subsets. Relative to YOLO11n, the full model raises mAP@0.5 by 0.0075 on Easy, 0.0080 on Hard, and 0.0135 on Hidden. Fixed recall increases by 0.0174, 0.0149, and 0.0245, respectively, while FN decreases by 147, 216, and 467. The largest absolute FN reduction and the largest fixed-recall improvement occur on Hidden, supporting consistent behavior on a second benchmark that includes deliberately concealed items. Because this experiment uses one split and training seed, the subset differences are descriptive and do not establish statistical significance.

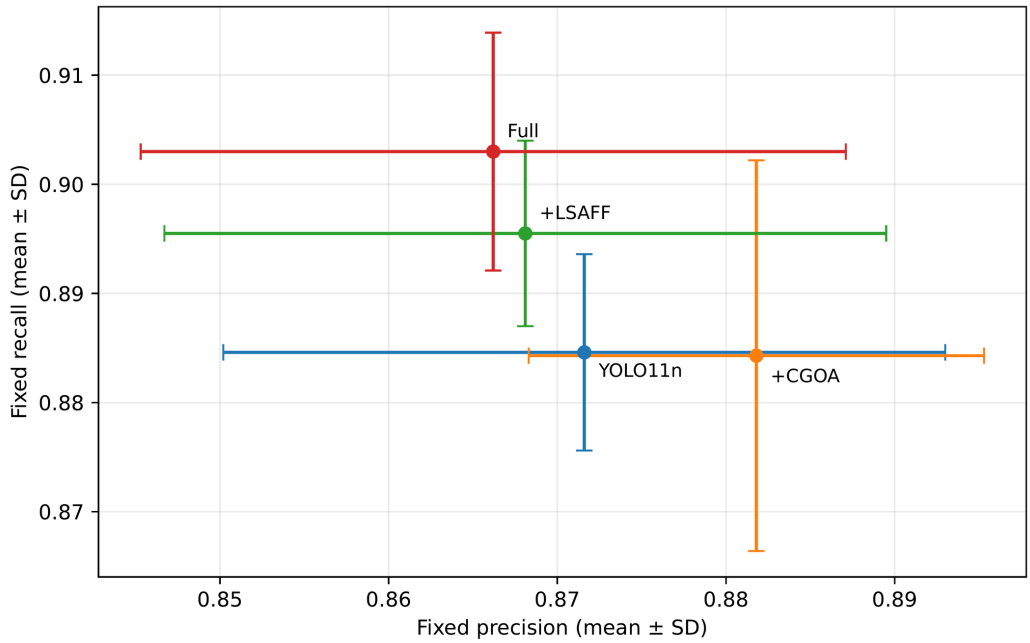


Fig. 8. Mean fixed precision and fixed recall over three split-and-training trials. Horizontal and vertical error bars show sample standard deviations.

TABLE X. SINGLE-SEED SUPPLEMENTAL SECOND-BENCHMARK VALIDATION ON THE COMBINED PIDRAY EASY, HARD, AND HIDDEN TEST SUBSETS. STANDARD P AND R ARE CLASS-AVERAGED VALIDATOR METRICS; FIXED P, FIXED R, AND F_2 ARE MICRO-AVERAGED AT CONFIDENCE 0.25 AND IOU 0.50

Model	P	R	mAP@0.5	mAP@0.5:0.95	TP	FP	FN	Fixed P	Fixed R	F_2
YOLO11n	0.7832	0.7498	0.7938	0.6582	31,852	9,373	10,148	0.7726	0.7584	0.7612
YOLO11n + LSAFF	0.7858	0.7621	0.8012	0.6675	32,368	9,502	9,632	**0.7731**	0.7707	0.7712
XraySafe-YOLO	**0.7885**	**0.7674**	**0.8035**	**0.6698**	**32,682**	9,782	**9,318**	0.7696	**0.7781**	**0.7764**

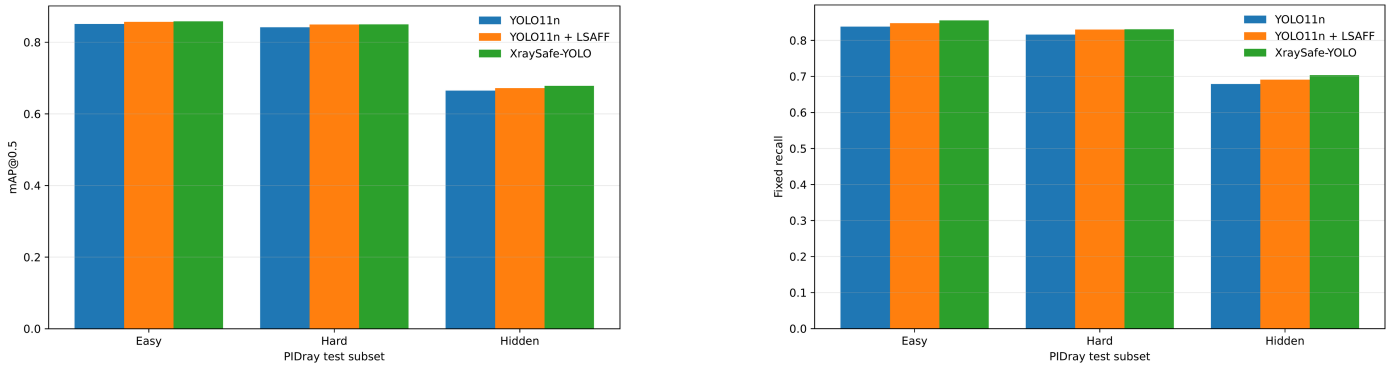


Fig. 9. Single-seed PIDray results across the official Easy, Hard, and Hidden subsets: (a) mAP@0.5 and (b) fixed recall at confidence 0.25 and IoU 0.50. The three subsets contain different numbers of ground-truth boxes; the normalized metrics are therefore more suitable than raw FN counts for cross-subset visual comparison.

E. Confidence Sweep and Operating-Point Interpretation

To test whether the recall advantage is only an artifact of applying the same confidence threshold to differently calibrated models, the saved seed-42 checkpoints were re-evaluated at confidence thresholds 0.10, 0.20, 0.25, 0.30, 0.40, and 0.50. Fig. 10 plots fixed recall against false positives per image for YOLO11n, LSAFF, and the full model. The full model remains above the baseline over most of the measured false-positive range, while LSAFF is intermediate. This post-hoc sweep uses the test set only to assess operating-point fairness; it is not used to choose a recommended deployment threshold. Any operational threshold should be selected on validation data from the target scanner and workload.

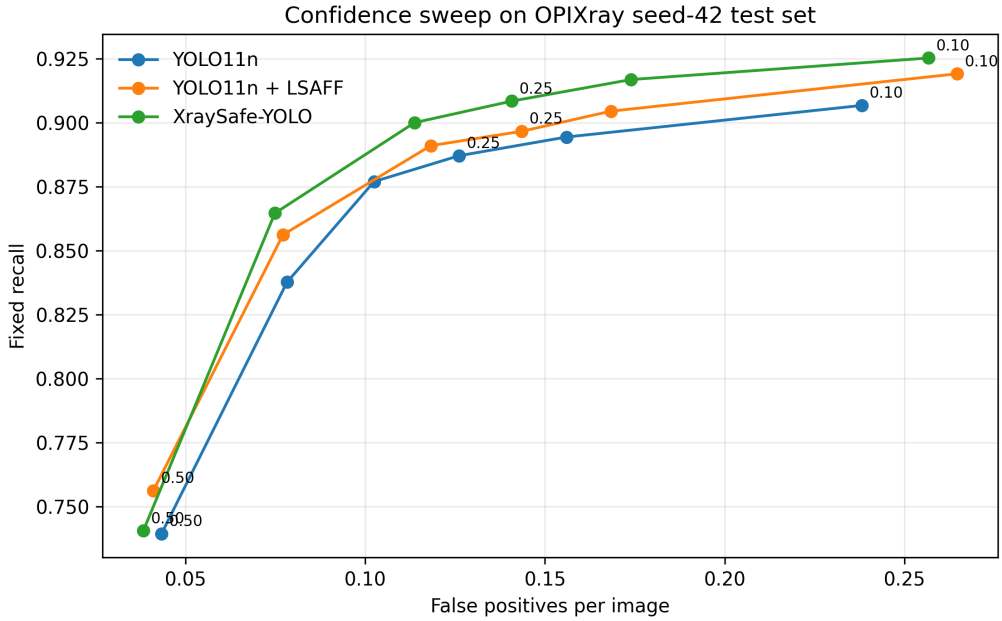


Fig. 10. Seed-42 OPIXray confidence sweep for YOLO11n, YOLO11n + LSAFF, and the full model. Recall is plotted against false positives per image, which avoids assuming that identical confidence thresholds imply equivalent operating conditions. Labels mark selected confidence thresholds.

TABLE XI. SEED-42 OPIXRAY RECALL AT MATCHED FALSE-POSITIVE BUDGETS, OBTAINED BY LINEAR INTERPOLATION BETWEEN ADJACENT MEASURED CONFIDENCE-SWEEP POINTS

Model	150 FP (0.0845 FP/image)	200 FP (0.1126 FP/image)	250 FP (0.1408 FP/image)
YOLO11n	0.8477	0.8813	0.8907
YOLO11n + LSAFF	0.8624	0.8862	0.8960
XraySafe-YOLO	**0.8733**	**0.8990**	**0.9084**

TABLE XII. MODEL COMPLEXITY AT 640-PIXEL INPUT RESOLUTION

Model	Params (M)	GFLOPs	Model Size (MB)
YOLO11n	2.583	6.317	5.216
YOLO11n + CBAM-lite	2.587	6.324	5.231
YOLO11n + CGOA	2.589	6.339	5.238
YOLO11n + LSAFF	2.893	7.227	5.877
YOLO11n + CGOA + LSAFF	2.899	7.249	5.898

For a matched-false-positive view, recall was linearly interpolated between adjacent measured sweep points at common FP budgets of 150, 200, and 250 detections over the 1776-image OPIXray test set. Table XI shows that the full model has the highest recall at all three budgets. At 200 FP, for example, the interpolated recalls are 0.8813 for YOLO11n, 0.8862 for LSAFF, and 0.8990 for the full model. Because these values are interpolated summaries of the measured sweep rather than independently optimized thresholds, they are used only as descriptive matched-operating-point evidence.

The fixed-count comparison at confidence 0.25 and IoU 0.50 remains useful as a common diagnostic point. At this threshold, the combined model reduces FN and increases FP relative to YOLO11n on both OPIXray and PIDray. The OPIXray three-trial trade-off in Fig. 8 indicates that this behavior is not limited to the seed-42 count table: the full model has the highest mean fixed recall and lower mean fixed precision than the baseline. The single-seed PIDray result shows the same direction on a broader category set. The sweep and matched-FP analysis strengthen the interpretation that the architectural change provides recall benefit beyond simply lowering the baseline detector's confidence threshold, while the final operating threshold remains an application-specific validation choice.

F. Model Complexity and Qualitative Examples

Table XII reports parameter count, GFLOPs, and model size at 640-pixel input resolution. The combined model has 2.899 M parameters and 7.249 GFLOPs, about 12.2% more parameters and 14.8% more computation than YOLO11n. Latency and FPS are not compared because they depend strongly on the local hardware and software environment.

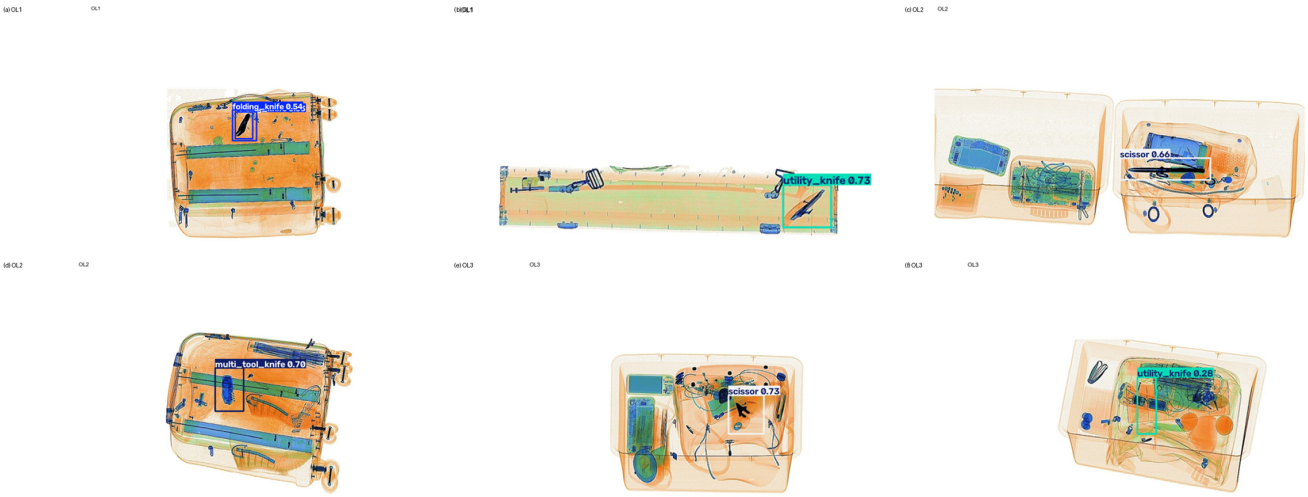


Fig. 11. Representative seed-42 predictions from the combined model: (a,b) OL1, (c,d) OL2, and (e,f) OL3.

Fig. 11 presents six combined-model predictions spanning OL1–OL3. Panels (a) and (b) show compact and elongated cutters whose visible boundaries remain distinguishable from the surrounding baggage structure. Panels (c) and (d) illustrate localization when a prohibited item overlaps dense electronics or elongated metal objects. Panels (e) and (f) are drawn from OL3; the utility-knife prediction in panel (f) has confidence 0.28, close to the fixed-count threshold, and illustrates the narrow margin available in a heavily cluttered scene. The examples show that the model can produce usable localizations at each occlusion level, but they are not a paired baseline comparison. Relative conclusions therefore rely on the complete test-set metrics and error counts rather than on selected images.

VI. Discussion

A. What the Experiments Show

The two modules have different effects. LSAFF is the more stable individual module. Across the three trials, it improves mean mAP@0.5, reduces mean FN, and increases mean fixed recall. This is reasonable for OPIXray because the detector needs both high-resolution details from partly visible blades and wider context from nearby objects. LSAFF lets each detection scale use information from P3, P4, and P5.

CGOA is less stable. Boundary-motivated local reweighting may help a partly visible item, but X-ray baggage also contains many strong edges from wires, frames, and overlapping materials. The CGOA model is below the baseline at seed 42 and has the largest variation across the three trials. Its mean FN is almost the same as YOLO11n. When the CGOA-enhanced P3 and P4 features are fused with P5 by LSAFF, the full model gives a lower mean FN than LSAFF alone. These results suggest that CGOA is useful mainly as a complement to multi-scale fusion, not as a better standalone attention module.

The class and occlusion results also show that the gain is not uniform. Multi-tool and utility knives improve, folding-knife recall decreases slightly, and straight knife remains difficult. The full model has the lowest FN on OL1 and OL3, while LSAFF is best on OL2. The clearest benefit of the full model is therefore fewer missed detections at the selected threshold.

The PIDray result shows a similar pattern on 12 classes. LSAFF improves all reported aggregate metrics and keeps fixed precision close to the baseline. The full model further reduces FN on Easy, Hard, and Hidden, with the largest fixed-recall gain on Hidden. FP also increases, so the PIDray result supports the same recall-oriented trade-off rather than a general improvement in every metric.

B. Relation to Recent X-ray YOLO Research

Recent compact X-ray detectors often combine several improvements and test them on more than one dataset. Examples include channel reduction with BiFPN [15], feature-aggregation redesign with normalized Wasserstein distance [16], state-space blocks evaluated on three datasets [21], contrastive multi-scale learning [22], and sparsity with adaptive distillation [23]. Those studies provide broader evidence on multiple runs, datasets, or inference speed than this study.

XraySafe-YOLO uses repeated OPIXray ablations and a single-seed PIDray second-benchmark test. It keeps the YOLO11n baseline small, separates the proposed modules, and exposes the fixed-threshold TP/FP/FN trade-off on both benchmarks. This is useful because a higher mAP alone does not show how many missed items are recovered at a selected threshold. At the same time, the PIDray result is not a state-of-the-art benchmark study and should not be compared directly with models trained under different splits, schedules, resolutions, or augmentation protocols.

C. Operational Meaning of the FP–FN Trade-off

At confidence 0.25, the seed-42 OPIXray full model finds 38 additional ground-truth objects and produces 26 additional false positives relative to YOLO11n, equivalent to 1.46 recovered objects per added false alarm. Across all three OPIXray trials, it lowers FN and raises fixed recall relative to the corresponding YOLO11n run, while fixed precision remains

slightly lower. PIDray shows the same direction: the full model recovers 830 additional ground-truth instances and produces 409 additional false positives, or 2.03 recovered objects per added false alarm. These ratios only describe the observed counts. A missed dangerous item and an unnecessary review do not have the same operational cost. In a human-in-the-loop screening system, the observed shift may be reasonable when the cost of a missed prohibited item is greater than the cost of secondary inspection. It is not automatically preferable, however. Additional false alarms consume operator time, can slow baggage throughput, and may reduce trust if the workload becomes excessive.

The preferred threshold depends on staffing, scanner type, traffic volume, and the relative cost of missed items and false alarms. The fixed threshold in this study is a common diagnostic point. Before deployment, threshold selection should use data from the target scanner, include calibration and confidence-sweep analysis, and report recall at a matched false-positive rate or operational workload.

D. Efficiency and Practical Use

The full model remains compact, with 2.899 M parameters and 7.249 GFLOPs at an input size of 640 pixels. Its parameter count is 12.2% above YOLO11n and its theoretical computation is 14.8% higher. These results show that the added modules have limited structural overhead. They do not prove real-time speed or hardware efficiency. FLOPs do not replace measured latency, and recent X-ray studies that use deployment-oriented language also report FPS or hardware-specific inference time [21], [23]. Accordingly, the present study uses "compact" for the detector as a whole and uses "lightweight" only for individual low-overhead operations or modules, and when describing related work that adopts that design objective.

The detector is intended to mark candidate regions for an inspector. It is not designed to make an automatic release decision. The operating threshold, escalation rules, and review interface would need to be evaluated together with detector accuracy in an operational study.

E. Limitations and Future Work

The primary repeated-trial evidence remains limited to OPIXray and its five cutter-like categories. PIDray adds a broader 12-class second-benchmark test with official Easy, Hard, and Hidden subsets, but it uses only one validation split and training seed and compares three configurations rather than the complete ablation. It therefore provides useful second-benchmark evidence without establishing robustness across repeated PIDray runs, HiXray, CLCXray, dual-view systems, other scanner manufacturers, cargo images, or operational customs data. Several recent models report results on two or three datasets [15], [21]–[23], so broader cross-domain evaluation remains important.

The repeated evaluation includes only three trials and changes the validation split together with the training seed. The reported standard deviations therefore combine split and optimization variability and are not fixed-split significance estimates. The overlapping error bars in Fig. 7 and 8 should be interpreted accordingly. A stronger design would keep the validation set fixed, repeat every configuration with more seeds, and apply paired bootstrap analysis to saved test predictions.

Other limitations concern experimental coverage. The automatic optimizer policy can couple architecture choice with automatically selected optimizer settings. The generic attention comparison is limited to CBAM-lite, and ASFF or BiFPN were not trained under the same protocol. CGOA alone was not repeated on PIDray. No fixed-split significance test, image-level bootstrap interval, measured latency, direct fusion-baseline training, attention-map analysis, or paired baseline qualitative panel is reported. For this reason, the manuscript does not claim real-time performance. CGOA uses no contour supervision and remains unstable as a standalone module on OPIXray. Very small, low-contrast, heavily overlapped, and visually ambiguous objects also remain difficult.

Future work will prioritize multi-seed second-benchmark validation on additional datasets, fixed-split repeated training, direct fusion baselines, confidence calibration, measured inference speed, and paired visual error analysis. A human-in-the-loop study should then quantify whether the reduction in missed detections justifies the additional secondary-inspection workload.

VII. Conclusions

XraySafe-YOLO combines scale-weighted fusion and contour-motivated attention in a compact YOLO11n detector. On OPIXray, LSAFF is the stronger standalone module, while CGOA is more sensitive to the validation split and training seed. In all three trials, the full model reduces FN and increases fixed recall compared with YOLO11n. It has the lowest mean false-negative count (172.7 ± 19.4) and the highest mean fixed recall (0.9030 ± 0.0109). At seed 42, FN decreases from 201 to 163 and fixed F_2 increases from 0.8848 to 0.8996, while FP increases from 224 to 250. The class and occlusion results show that the improvement is useful but not the same for every case.

The PIDray experiment shows the same main direction on 12 classes. mAP@0.5 increases from 0.7938 to 0.8035, FN decreases from 10,148 to 9,318, and fixed recall increases from 0.7584 to 0.7781. Fixed precision decreases slightly from 0.7726 to 0.7696. The largest fixed-recall gain is on the Hidden subset. Because PIDray is tested with one seed, this result supports the architecture on a second dataset but does not replace repeated second-benchmark validation.

The model should be viewed as a compact recall-oriented option for human-assisted X-ray screening, not as a detector that is better in every metric or ready for direct deployment. The main evidence comes from the fixed-threshold error analysis, repeated OPIXray trials, and the PIDray second-benchmark test. Stronger conclusions would require repeated tests on more datasets and scanners, direct fusion baselines, threshold calibration, and measured inference speed.

Acknowledgment

The authors thank the OPIXray and PIDray maintainers for making the benchmarks available to the research community.

DECLARATIONS

Funding: This research was funded by the Scientific Research Startup Fund of Shanghai Customs College under Grant No. 2315053A2024.

Data availability: The experiments use the public OPIXray and PIDray benchmarks. The original images and annotations are not redistributed and should be obtained from the official repositories at https://github.com/OPIXray-author/OPIXray and https://github.com/lutao2021/PIDray, subject to their academic-use terms. Processed result summaries, model-structure code, configuration files, and demonstration materials supporting the findings are available from the corresponding author upon reasonable request.

Conflict of interest: The authors declare no conflict of interest.

REFERENCES

[1] C. Miao, L. Xie, F. Wan, S. Su, H. Liu, J. Jiao, and Q. Ye, "SIXray: A large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2119–2128.

[2] Y. Wei, R. Tao, Z. Wu, Y. Ma, L. Zhang, and X. Liu, "Occluded prohibited items detection: An X-ray security inspection benchmark and de-occlusion attention module," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 138–146, doi: 10.1145/3394171.3413828.

[3] OPIXray Authors, "OPIXray: Occluded prohibited items X-ray dataset." [Online]. Available: https://github.com/OPIXray-author/OPIXray. Accessed: Aug. 18, 2026.

[4] B. Wang, L. Zhang, L. Wen, X. Liu, and Y. Wu, "Towards real-world prohibited item detection: A large-scale X-ray benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 5412–5421.

[5] L. Zhang, L. Jiang, R. Ji, and H. Fan, "PIDray: A large-scale X-ray benchmark for real-world prohibited item detection," *Int. J. Comput. Vis.*, vol. 131, pp. 3170–3192, 2023, doi: 10.1007/s11263-023-01855-1.

[6] L. Zhang, L. Jiang, R. Ji, and H. Fan, "PIDray: A large-scale X-ray benchmark for real-world prohibited item detection." [Online]. Available: https://github.com/lutao2021/PIDray. Accessed: Aug. 18, 2026.

[7] M. Li, T. Jia, H. Wang, B. Ma, H. Lu, S. Lin, D. Cai, and D. Chen, "AO-DETR: Anti-overlapping DETR for X-ray prohibited items detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, pp. 12076–12090, 2025, doi: 10.1109/TNNLS.2024.3487833.

[8] R. Tao, H. Wang, Y. Guo, H. Chen, L. Zhang, X. Liu, Y. Wei, and Y. Zhao, "Dual-view X-ray detection: Can AI detect prohibited items from dual-view X-ray images like humans?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 10338–10347.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[12] Ultralytics, "YOLOv8 documentation." [Online]. Available: https://docs.ultralytics.com/. Accessed: Aug. 18, 2026.

[13] Ultralytics, "YOLO11 documentation." [Online]. Available: https://docs.ultralytics.com/models/yolo11/. Accessed: Aug. 18, 2026.

[14] X. Huang and Y. Zhang, "ScanGuard-YOLO: Enhancing X-ray prohibited item detection with significant performance gains," *Sensors*, vol. 24, Art. no. 102, 2024, doi: 10.3390/s24010102.

[15] Y. Fang, C. Xu, and Y. Zhang, "Research on X-ray security contraband identification technology based on lightweight YOLOv8," *Sci. Rep.*, vol. 14, Art. no. 25031, 2024, doi: 10.1038/s41598-024-75932-x.

[16] Y.-T. Zhou, K.-Y. Cao, D. Li, and J.-C. Piao, "Fine-YOLO: A simplified X-ray prohibited object detection network based on feature aggregation and normalized Wasserstein distance," *Sensors*, vol. 24, Art. no. 3588, 2024, doi: 10.3390/s24113588.

[17] H. Zhang, Z. Zhao, and J. Yang, "Attention-based prohibited item detection in X-ray images during security checking," *IET Image Process.*, vol. 18, pp. 1119–1131, 2024, doi: 10.1049/ipr2.13013.

[18] H. Zhang, W. Teng, X. He, H. Que, and Y. Zhang, "Lightweight prohibited items detection model in X-ray images based on improved YOLOv7-Tiny," *J. Franklin Inst.*, vol. 362, Art. no. 107421, 2025, doi: 10.1016/j.jfranklin.2024.107421.

[19] F. Guan, H. Zhang, and X. Wang, "An improved YOLOv8 model for prohibited item detection with deformable convolution and dynamic head," *J. Real-Time Image Process.*, vol. 22, Art. no. 84, 2025, doi: 10.1007/s11554-025-01665-3.

[20] J. Chen, J. Hao, and X. Liu, "An X-ray contraband detection method based on improved YOLOv8," *IET Image Process.*, vol. 19, Art. no. e70135, 2025, doi: 10.1049/ipr2.70135.

[21] K. Zhao, S. Peng, Y. Li, and T. Lu, "A lightweight Xray-YOLO-Mamba model for prohibited item detection in X-ray images using selective state space models," *Sci. Rep.*, vol. 15, Art. no. 13171, 2025, doi: 10.1038/s41598-025-96035-1.

[22] Q. Diao, W. Chan, A. M. Zain, K. Moorthy, Y. Chen, H. Tong, and Y. Huo, "A lightweight multi-scale detection framework for X-ray images with supervised contrastive learning," *Sci. Rep.*, vol. 16, Art. no. 8635, 2026, doi: 10.1038/s41598-026-38000-0.

[23] J. Kong, "GSA-YOLO enables high-efficiency real-time X-ray security inspection via structured sparsity and adaptive knowledge distillation," *Sci. Rep.*, advance online publication, 2026, doi: 10.1038/s41598-026-55599-2.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.

[26] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[27] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13713–13722.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2117–2125.

[29] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8759–8768.

[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10781–10790.

[31] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," *arXiv preprint arXiv:1911.09516*, 2019.

[32] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21002–21012.

Individualizing an Agent-Based Flood Evacuation Model from Survey Data

Tatsuki Fukuda
Faculty of Data Science, Shimonoseki City University, Shimonoseki, Japan

***Abstract*—Earlier agent-based models (ABMs) of flood evacuation by the author, first on an abstract grid and later extended to real 3D city-model geography, use a survey of Japanese residents to assign each simulated agent to one of a few discrete behavioral groups; within a group, the evacuation probability P and the neighbor-awareness radius L are, with one narrow exception, fixed at a single value shared by every agent. The same survey contains two individually answered questions – how much a respondent cares about neighbors' evacuation status, and how many neighboring houses' worth of information they want. The first was never used in that prior work, and the second only once, in isolation, on an abstract grid; both are natural candidates for individualizing P and L, respectively. This paper extends the model in four ways: whole respondent records are resampled, preserving the real correlation among a respondent's evacuation threshold, concern level, and preferred radius; the preferred-neighbor count is converted into a metric radius using the actual local building density from real PLATEAU 3D city-model data for Toyohashi City; the persistent Evacuate-Now-Button regime and a direct-observation regime are unified within one model; and a real-geometry video visualization is added. Re-deriving the group distribution from the survey matches the previously published values to within 1.5 percentage points, and the observation regime reproduces the prior "no-system" rate (9.6% vs. ≈9%) as an emergent consequence rather than a calibrated target; an ablation shows that individualizing P alone, under an assumed concern-to-probability mapping, shifts the final evacuation rate by nearly 18 points (79.9% vs. 62.0%). Throughout, the parameters that remain researcher-declared assumptions are distinguished from those derived from data.**

***Keywords*—Agent-based model; flood evacuation; evacuation rate; survey-based simulation; evacuate now button**

I. Introduction

Japan is exposed every year to typhoons and heavy-rain events, and despite the routine issuance of evacuation advisories, the fraction of at-risk residents who actually evacuate remains low. A nationwide survey found that even under a Level 4 alert (nominally an evacuation order), only 16.9% of residents at home evacuated to a location outside their home, and for hydro-meteorological hazards specifically the share taking evacuation action was only 5.7% [1]. In the June 2023 heavy rain event in Toyohashi City, Aichi Prefecture, which serves as the case study of this paper, the recorded evacuation rate was approximately 0.068% [2].

Several factors have been identified as contributing to this persistently low evacuation rate in Japan: many residents do not evacuate despite receiving official information [3]; personality traits such as being "cautious and proactive" rather than "careless and proactive" predict earlier action [4]; prior risk awareness promotes evacuation [5]; and the timing and channel of information delivery measurably affect uptake, as shown for the July 2018 Gifu Prefecture heavy rain [6] and for landslide evacuation [7]. Separately, methodological work has examined how to estimate flood evacuation rates from disaster records [8].

A body of prior work by the author has approached this problem with agent-based modeling (ABM), motivated by the observation that Japanese residents' evacuation decisions are strongly conditioned on whether their neighbors have evacuated [9]. This is consistent with the classic sociological threshold model of collective behavior [10], in which each individual has a personal threshold – the fraction of others who must already have acted before that individual acts – and the population-level dynamics follow from the distribution of thresholds across individuals; large-scale mobile-phone location data show a similarly sharp threshold pattern in real evacuation behavior after earthquakes, with evacuation probability rising sharply once measured seismic intensity crosses a critical level [11]. Building on the "Evacuate Now Button" concept, first proposed in an earlier technical report [12] – a physical button pressed at each household to share residents' evacuation status – prior work [9] used a first ABM prototype, on an abstract (non-geographic) grid, to explore how the number of neighbors whose status is shared affects the resulting evacuation rate. This was substantially extended [13] with real PLATEAU 3D city-model geography, an explicit warning-level schedule (Levels 3/4/5) reproducing a real disaster timeline, and a two-group behavioral model (an alert-only Group I and a neighbor-rate-triggered Group II, further split into five threshold sub-groups) calibrated from the same survey of 1,000 respondents used in [9]. The Evacuate Now Button concept was then implemented [14] as a working IoT prototype and its practical feasibility measured (end-to-end latency, delivery reliability under simultaneous presses, battery life), providing empirical grounding for the assumption – used throughout the ABM line of work, including this paper – that a button-based system can share evacuation status quickly and reliably.

This paper's contribution is to make fuller use of the same underlying survey than these two prior papers [9], [13] did. The later extension [13] uses the survey only to assign discrete threshold groups, applying a single assumed evacuation probability ($P = 0.5$, not itself estimated from the survey) uniformly to every agent; [9]'s earlier prototype instead set separate, survey-derived probabilities for its alert-only and neighbor-threshold groups, but neither prior paper varies P *within* a group by an individual respondent's stated concern, which is

the specific individualization Q7 enables here. Both papers also use, in almost every experiment, a single, population-wide neighbor-awareness parameter L (swept as an experimental condition in [13]; individualized as a per-agent neighbor *count*, but only in one isolated experiment of [9], on an abstract, non-geographic grid). The survey, however, also contains two further single-select questions – one asking how much a respondent cares about neighbors' evacuation status, which neither prior paper uses at all, and one asking how many neighboring houses' worth of information they would want, which the earlier prototype [9] uses only in that one isolated experiment. These two questions are used here to individualize, respectively, the evacuation probability P and the awareness radius L for each simulated agent; resampling whole respondent records (rather than each question's marginal distribution independently) preserves, by construction, the real correlation between an agent's threshold sensitivity, stated concern, and preferred information radius. The "information regime" an agent operates under – a persistent, always-accurate Evacuate-Now-Button-style regime on one hand, and a regime with no such system, in which agents can only directly notice a recent, nearby evacuation, on the other – is further generalized into a single switchable parameter of the same individualized model, rather than two separately implemented mechanisms. Finally, a visualization pipeline is added that renders one stochastic realization of the simulation directly onto real PLATEAU building footprints as a video, with a live numeric summary panel.

Throughout, a careful distinction is drawn between two kinds of improvement over prior work: 1) replacing a previously unfounded or population-wide assumption with a quantity estimated directly from previously unused survey data, and 2) methodological or software-engineering improvements (reproducibility, open implementation, explicit separation of data-derived and assumed parameters) that do not by themselves improve predictive accuracy. Several parameters – the base evacuation probability, the per-evaluation discount rate, and the direct-observation range and recency window – remain researcher-declared assumptions inherited from [13], because the survey contains no question from which they could be estimated, together with the CareScale mapping newly introduced here (Section IV-A); this is reported explicitly rather than presenting the whole model as empirically calibrated.

The remainder of this paper is organized as follows. Section II reviews the three prior papers this work extends and situates the contribution relative to each. Section III describes the survey and the derivation of the group-threshold distribution. Section IV presents the individualization of P and L. Section V describes the two information regimes. Section VI describes the real-geography case study. Section VII gives the simulation algorithm. Section VIII reports experimental results. Section IX discusses the findings, and Section X states limitations, followed by a conclusion in Section XI.

II. Related Work

Beyond the three papers this work directly extends, ABM, a general methodology for simulating heterogeneous individuals with simple behavioral rules [15], has been applied extensively to flood evacuation and related behavior, including social-media and individual influences on evacuation decisions [16], flood-event inhabitant behavior [17], mutual-aid mechanisms in vulnerable communities [18], flash-flood warning response [19], flood-pedestrian interaction dynamics [20], [21], flood-induced displacement forecasting [22], community flood-risk mitigation [23], coastal migration under future flood risk [24], and household adoption of private flood-mitigation measures [25]; see [26] for a broader review of agent-based flood risk models. To the author's knowledge, none of these models individualizes both an evacuation probability and an awareness radius from the same joint survey record, or unifies a persistent-sharing and a direct-observation information regime within one model, as is done here.

A parallel strand models the physical dynamics of pedestrian and crowd evacuation rather than the decision to evacuate: cellular-automaton and continuous crowd models capture movement effects such as turning behavior on clearance time [27], controlled experiments have measured human motion speed in floodwaters [28], and recent reviews survey a decade of progress in agent-based and pedestrian evacuation simulation [29], [30]. The model developed here deliberately abstracts away physical movement – representing only the evacuate-or-stay decision and its timing – so this body of work is complementary to, rather than in competition with, the present approach.

Most closely related is work on social influence in evacuation, since the agents' core mechanism is neighbor-conforming threshold behavior [10]. Imitation of cooperative behavior has been incorporated into pedestrian evacuation models and shown to reshape clearance dynamics [31]. At the individual level, evacuation decisions have been predicted from pre-disaster web-search behavior [32] and from large-scale mobile-phone location traces [11], indicating that heterogeneity in individual evacuation propensity is both real and, to a degree, measurable – which is precisely the heterogeneity that the survey-based individualization of P and L presented here aims to capture.

An initial ABM prototype based on the Evacuate Now Button concept, first proposed in [12], was presented in earlier work [9], which framed the evacuate-or-stay decision using game theory [33] and grounded the neighbor-conforming behavior in bandwagon-effect theory [34]. Agents were placed at random screen coordinates (no real geography). A survey of 1,000 respondents asked, among other things, which of several conditions – official alert levels near home or in a neighboring town, or five neighbor-evacuation-rate levels (10/30/50/80/almost-all%), or "I don't care about others" – would make the respondent evacuate; this is the same question used in Section III. Ninety percent of agents were assigned to a neighbor-rate-driven behavior and 10% to an alert-only behavior; both groups' probabilities were set directly from this same survey question rather than assumed – the alert-only group's from the official-alert responses (18.6% at Level 3, 79.6% at Level 4), and the neighbor-rate-driven group's from the corresponding neighbor-rate responses (e.g., 50% of respondents said they would evacuate once half of their neighbors had). A separate survey question asked how many neighboring houses' status a respondent would want to know (10/25/50/100/200); in one experimental condition, this distribution (46%/29%/15%/7%/3%, respectively) was used to assign each agent an individual *count* of neighbors to observe,

though on the abstract, non-geographic grid this count carried no notion of physical distance, and it was tested only as one alternative to several fixed values (0, 10, 20, 50, 100, 200), not integrated with any other individual-level variation. (This split is the raw Q11 marginal reported in Section III – 46.0/29.1/15.0/6.6/3.3% – rounded to integer percentages, confirming that the earlier prototype [9] drew on the same question.) This question is built on directly in Section IV-B, where it is converted into a metric radius using real building density and combined, via joint resampling (Section IV-C), with the same agent's threshold group and stated concern level.

A subsequent extension replaced the abstract grid with real PLATEAU 3D city-model geography (Nobeoka City and Toyohashi City) and a real warning-level timeline reproducing an actual disaster [13]. Agents were split into an alert-only Group I (≈8% of respondents) and a neighbor-rate-driven Group II, itself split into five threshold sub-groups (10/30/50/80/100%, in proportions 20%/31%/32%/12%/5%, Table I therein). Every agent used the same fixed evacuation probability ($P = 0.5$) and the same awareness radius L within a given experiment, with L swept from 20 to 100 m across experiments to study its effect on the final evacuation rate, and a separate, simplified "without system" baseline (observing only evacuations within 40 m and the last 5 minutes, with no threshold-group structure) was implemented once for comparison. Keeping P and L population-wide fit that design's aim – a controlled sweep of one macro-level parameter, not a search for individual-level heterogeneity – and neither the radius conversion of Eq. (2) nor the joint-resampling of Section IV-C, both introduced here, had yet been applied there to turn Q11 and Q7 into individualized, mutually consistent parameters. In the present paper, the Group I/II(a–e) proportions are reproduced independently from the raw survey data (Section III); both P and L are individualized per agent instead of using one shared value per experiment (Section IV); and the "without system" baseline is re-implemented as one setting of a single, unified information-regime parameter that uses the same per-agent threshold groups as the main model (Section V), rather than as a separate, simplified mechanism.

A companion hardware study implemented the Evacuate Now Button as a working prototype (ESP-WROOM-02 plus a cloud pipeline) and measured end-to-end latency (median 1.9 s, 2.3 s under eight near-simultaneous presses), delivery reliability (zero losses in 200 simultaneous presses, 95% upper bound on loss rate 1.5%), and battery life (>30 months under several usage scenarios) [14]. That study does not modify the simulation model, but its results directly support the *button* regime of Section V: the regime's core assumption – that a button press becomes known, essentially instantaneously and reliably, to a shared aggregation service – is technically achievable, not merely theoretical.

III. Survey and the Group-Threshold Distribution

The survey (Lancers online panel, $n = 1000$, ages spanning teens to 80s) is the same questionnaire referenced in [9], [13]. This paper uses three of its single- or multi-select items:

- **Q6** (multi-select checkbox): "If the following situation occurred while you were at home during heavy rain, would you evacuate to a shelter? Select all that you think would make you evacuate." Eleven options mix five official-alert conditions (not used here) with five neighbor-evacuation-rate levels (10/30/50/80%, almost all) and an "I don't care about others" option.

- **Q7** (single-select): "When deciding whether to evacuate during heavy rain, do you care whether the houses around yours have evacuated?" – *cares* (501), *cares a little* (420), *doesn't care at all* (79).

- **Q11** (single-select): "If you were to refer to how many neighboring houses have evacuated, how many would you want data for?" – 10 (460), 25 (291), 50 (150), 100 (66), 200-or-more (33).

A. Group Classification Rule

Because Q6 allows multiple selections, a rule is needed that maps a respondent's checked options to a single effective threshold. A respondent is classified as Group I if they checked "I don't care about others" and none of the five numeric levels. If they checked at least one numeric level, the *lowest* checked level is taken as their effective threshold (Group II(a–e)): checking a higher level is consistent with also being willing to evacuate at a lower one, so the minimum checked level is the most informative single estimate of the respondent's true trigger point. Respondents who checked neither a numeric level nor "I don't care" (i.e., who only responded to the alert-based options, or left the relevant options blank) provide no information about a neighbor-rate threshold and are excluded from the distribution.

B. Independent Reproduction of a Prior Group-Threshold Distribution

Applying this rule to the raw data yields the distribution in Table I. Of 1,000 respondents, 758 (75.8%) are classifiable; the remaining 242 (24.2%) are excluded as described above. Table I of [13] reports Group I ≈8%, and, within Group II, sub-group proportions of 20%, 31%, 32%, 12%, and 5% for the 10%, 30%, 50%, 80%, and 100% thresholds – but does not describe how these figures were derived from the raw multi-select responses. The classification rule was not fit to these published numbers; the rule above was fixed on the multi-select data itself, independent of the target values. Computed on the same basis as [13] (Group I as a share of all classified respondents, and the Group II sub-groups as shares within Group II), the resulting distribution matches [13]'s Table I to within 1.5 percentage points (the largest single deviation is Group I, 9.5% vs. ≈8%; every Group II sub-group agrees to within 0.6 points). This agreement is reported as an independent check that the (fully specified, reproducible) rule used here is likely to recover the procedure used in [13], which was not documented there, rather than as a claim of numerical novelty. The 24.2% of respondents who provide no usable answer to the neighbor-rate item are not discussed in [13]; their exclusion, rather than assignment to Group I, is treated as an explicit, open modeling choice (Section X).

TABLE I. GROUP-THRESHOLD DISTRIBUTION DERIVED INDEPENDENTLY FROM THE RAW SURVEY DATA (n=758; GROUP II ROWS: n=686), COMPARED WITH [13]'S TABLE I. RAW COUNTS IN PARENTHESES

Group	Threshold	This work	[13]
Group I ("don't care")	—	9.5% (72)	≈8%
Group II(a)	10%	20.6% (141)	20%
Group II(b)	30%	30.5% (209)	31%
Group II(c)	50%	31.6% (217)	32%
Group II(d)	80%	12.0% (82)	12%
Group II(e)	almost all (100%)	5.4% (37)	5%

IV. INDIVIDUALIZING EVACUATION PROBABILITY AND AWARENESS RANGE

A. Evacuation Probability P via Q7

In [13], every agent uses the same evacuation probability P=0.5 at every eligible evaluation. P is instead scaled per agent, for Group II agents only, using that agent's bootstrapped answer to Q7:

$$P_{\text{agent}} = P_{\text{base}} \times \text{CareScale}(\text{Q7 answer}) \tag{1}$$

where, CareScale $\in \{1.0,\ 0.6,\ 0.25\}$ for *cares*, *cares a little*, and *doesn't care at all*, respectively. Group I agents are unaffected, since Q7 asks specifically about neighbor status, which Group I agents do not use. The three scale values in Eq. (1) are a declared modeling assumption, not a measured quantity: no survey question asks directly how likely a respondent is to act on their stated intention, so converting an ordinal concern level into a probability multiplier necessarily involves a researcher-chosen mapping, anchored at 1.0 for the "cares" response. What is new is the use of a previously unused survey item to introduce principled, data-linked individual variation in P in place of a single population-wide constant; the specific numeric mapping is a target for future calibration (Section X).

B. Awareness Radius L via Q11 and Real Building Density

The later extension [13] sweeps a single radius $L \in \{20, 40, 60, 80, 100\}$ m, shared by every agent, across separate experiments. The earlier prototype [9] individualizes a neighbor *count* per agent from the same Q11-style question, but on an abstract grid with no notion of metric distance. L is individualized here by converting each agent's bootstrapped Q11 answer (a preferred house count N_i) into a metric radius L_i using the real local building density ρ (households per m^2) of the loaded case-study dataset, assuming households are uniformly distributed at that density:

$$L_i = \sqrt{\frac{N_i}{\pi\rho}}. \tag{2}$$

For the Toyohashi case study of Section VI ($\rho \approx 1{,}752$ households/km^2), Eq. (2) maps the five Q11 answers to radii of 42.6, 67.4, 95.3, 134.8, and 190.6 m, respectively – comparable in magnitude to, but individually variable around, the single fixed values (20–100 m) swept in [13]. Table II gives the resulting per-agent radius distribution over the bootstrap pool: it is strongly weighted toward the two smallest radii (72.7% of the 758-respondent bootstrap pool at ≤67 m; median 67.4 m, mean ≈71 m), which is why the individualized-button result (62.0%) falls between the fixed L=60 m and L=80 m results (Section VIII-B).

TABLE II. PER-AGENT AWARENESS RADIUS INDUCED BY Q11 (EQ. (2)) OVER THE BOOTSTRAP POOL OF 758 CLASSIFIABLE RESPONDENTS

Q11 answer (houses)	Radius L_i [m]	Share of agents
10	42.6	41.7%
25	67.4	31.0%
50	95.3	16.4%
100	134.8	7.3%
200	190.6	3.7%

C. Joint Bootstrapping of Q6, Q7, and Q11

All three questions above are recorded for the same 758 respondents classifiable on Q6 (Q7 and Q11 are mandatory single-select fields answered by all 1,000 respondents). Rather than sampling each agent's group, care level, and preferred house count from three independent marginal distributions, whole respondent rows are resampled with replacement, so that whatever correlation exists in the real data among threshold sensitivity, stated concern, and preferred information depth is carried into the simulated population. Neither [9] nor [13] resamples joint respondent profiles in this way; both treat group assignment, P, and L as independent parameters (or population-wide constants).

V. INFORMATION REGIMES: PERSISTENT SHARING VS. DIRECT OBSERVATION

A Group II(a–e) agent becomes eligible to evacuate once "the evacuation rate of surrounding households" reaches its personal threshold, but what that phrase means depends on whether a button-style sharing system exists.

Let $\mathcal{N}_i = \{j \neq i : d_{ij} \leq L_i\}$ be the set of other households within agent i's radius, where d_{ij} is the metric distance between households. In both regimes below, a rate is defined as 0 for an agent whose radius contains no other household ($|\mathcal{N}_i| = 0$), avoiding division by zero; such a Group II agent then never becomes eligible (Group I is unaffected, since it never consults neighbor information – Section IV). This is rare under the individualized radii of Eq. (2) and the observation regime's fixed L_{obs}=40 m (at most 0.4%, Section VI), but reaches about 4% at the smallest fixed-L condition tested (L=20 m, Section VIII-A).

1) Button regime: As in [13], a management center aggregates every button press, so at every evaluation an agent knows the true, persistent ("stock") evacuation rate among other households within its radius L_i (Section IV-B):

$$r_i^{\text{stock}}(t) = 100 \times \frac{\left|\{j \in \mathcal{N}_i : \text{evac}_j(t)\}\right|}{|\mathcal{N}_i|}, \tag{3}$$

where, $\text{evac}_j(t)$ indicates that j has evacuated by time t. This is the regime whose real-world feasibility is supported by the latency and reliability measurements of [14].

2) Observation regime: With no sharing system, an agent can only directly notice a neighbor's evacuation while it is happening – e.g., seeing them leave through a window – within a limited range and only if the agent happens to check during that narrow window. This is modeled as follows: at every T_1 check, the agent computes a recent ("flow") rate over a fixed range L_{obs} (so $\mathcal{N}_i^{\text{obs}} = \{j \neq i : d_{ij} \leq L_{\text{obs}}\}$),

$$r_i^{\text{flow}}(t) = 100 \times \frac{\left|\{j \in \mathcal{N}_i^{\text{obs}} : t - W \leq \text{evac\_time}_j \leq t\}\right|}{|\mathcal{N}_i^{\text{obs}}|}, \tag{4}$$

where, evac_time_j is the time at which j evacuated, and compares this recent rate (rather than the persistent stock rate of the button regime) against the same Q6-derived personal threshold. Because r^{flow} counts only evacuations in the trailing W-minute window it is almost always far smaller than r^{stock}, which is the mechanism behind the two regimes' very different outcomes (Section VIII).

Following [13]'s "without system" baseline, L_{obs} is set to 40 m and W to 5 min; as with P_{base} and the discount rate α (Section VII), these two values are not estimable from the survey and are adopted, as declared assumptions, from that prior baseline rather than derived from data. Because L_{obs} is specifically about *direct observation range* rather than *preferred aggregated-information depth*, the Q11-derived individualized radius is not reused here; Q11 presupposes a sharing system, which is precisely what the observation regime lacks.

Group I agents are unaffected by the choice of regime, since by construction they decide "on their own," independent of neighbor information, in both cases.

VI. Case Study: Real Building Geometry and a Real Warning Timeline

The same case-study city as [13], Toyohashi City, Aichi Prefecture, is used here for two reasons: it lets the present results be read alongside the prior paper's, and its June 2023 heavy-rain event provides real Warning Level 3/4/5 issuance times (06:40, 14:40, and 16:30 on June 2, 2023, respectively [2], [13]) that are reused as the simulation's warning schedule, rather than an arbitrary synthetic timeline.

Building positions are obtained directly from Project PLATEAU's public FY2023 CityGML package for Toyohashi [35], distributed under CC-BY 4.0. Only the needed building-model mesh tiles (≈1 km × 1 km JIS X0410 tertiary mesh cells) are fetched via HTTP range requests against the remote ZIP's central directory, avoiding the full ≈850 MB city-wide archive. Of the city's 288 such tiles, the one with the largest residential building count is used, mesh 52371229 (centered near 34.77°N, 137.37°E, in central Toyohashi), which contains 1,856 residential buildings (`bldg:usage` codes 411/412/413/414/415, i.e. detached houses, apartment buildings, and mixed-use residential buildings) within a bounding box of approximately 1.06 km^2, giving a local density $\rho \approx 1{,}752$ households/km^2. Each building's centroid and footprint (LOD0 roof-edge geometry, with a small placeholder square for the minority lacking it) are projected to local metric coordinates via an equirectangular approximation, adequate at this sub-10 km scale. One agent is instantiated per residential building, following the same convention as [13]; each building – including apartment and mixed-use buildings that may house multiple households – is therefore treated as a single household for the purposes of ρ and Eq. (2). Computed directly from this building layout, the fraction of households with zero neighbors within range is at most 0.4% for the individualized Q11-derived radii of Table II (minimum 42.6 m) and for the observation regime's fixed L_{obs}=40 m.

VII. Simulator

The simulator is implemented in Python (NumPy/pandas for the model; lxml/Shapely for geometry parsing), in contrast to the Unity/C# implementation of [13]. Agents are represented as parallel NumPy arrays rather than individual objects, since every per-tick operation is naturally vectorized across all agents due for evaluation at that tick.

1) Timing: As in [13], each agent re-evaluates whether to evacuate every $T_1 \sim U(7.5, 22.5)$ min (15 min ±50%), where $U(a, b)$ denotes the uniform distribution on the interval $[a, b]$, after an initial random delay $T_2 \sim U(15, 45)$ min (30 min ±50%) from the start of the simulation. The simulation clock advances in discrete steps of $\Delta t = 1$ min, beginning at Warning Level 3 (t=0) and ending at Level 5 (t=T_{end}, i.e. 590 min later). Level 4's issuance time (Table III) is shown on the temporal figures for reference only; it does not itself alter agent behavior in Algorithm 1.

2) Decision rule: At each eligible evaluation, a Group I agent is always eligible once the simulation has started; a Group II(a–e) agent is eligible only if the neighbor rate (Section V) meets or exceeds its personal threshold. An eligible agent evacuates with probability P_{agent} (Eq. (1)); if it does not, its probability is discounted in place for its next eligible evaluation, $P_{\text{agent}} \leftarrow \alpha P_{\text{agent}}$ (the multiplicative discount of [13], Eq. (1)), with α=0.8. Both P_{base}=0.5 and α=0.8 are carried over unchanged from [13] as declared assumptions, since the survey contains no question from which either could be estimated.

3) Neighbor-rate computation: At each tick, for every agent due for evaluation, pairwise distances to all other agents are computed and thresholded at that agent's own radius (individualized, Eq. (2), in the button regime; fixed at L_{obs} in the observation regime) to obtain the relevant neighbor set, from which the persistent or recent evacuation rate is computed as described in Section V. Algorithm 1 summarizes one stochastic run.

Table III consolidates every model parameter together with its value and its provenance. This is deliberate: a recurring theme of this paper is that only some of the model is estimated from data, and the table makes the boundary explicit at a glance. Two quantities – the Group distribution and the per-agent radius – are estimated directly from survey questions Q6 and Q11, respectively; Q7 supplies each agent's concern level, which is then converted into a per-agent probability multiplier via the assumed CareScale mapping (Eq. (1)); the local building density and the warning-issuance times come from real PLATEAU and disaster-record data; and the remaining five (P_{base}, α, the CareScale mapping, L_{obs}, W) are declared assumptions – four carried over unchanged from [13], and the CareScale mapping newly introduced in this paper – none of

TABLE III. MODEL PARAMETERS AND PROVENANCE. "SURVEY" = FROM THE QUESTIONNAIRE; "REAL DATA" = PLATEAU OR DISASTER RECORDS; "ASSUMED" = RESEARCHER-DECLARED; [13] = TIMING PARAMETERS UNCHANGED FROM THAT PAPER

Symbol	Meaning	Value	Provenance
Group dist.	Group I / II(a–e) shares	Table I	Survey (Q6)
CareScale	concern $\rightarrow P$ multiplier	$\{1.0, 0.6, 0.25\}$	Assumed
P_{base}	base evac. probability	0.5	Assumed
α	per-eval. discount	0.8	Assumed
T_1	re-evaluation interval	15 min $\pm$50%	[13]
T_2	initial delay	30 min $\pm$50%	[13]
L	awareness radius (button)	per agent, Eq. (2)	Survey (Q11)
ρ	local building density	1,752 /km^2	Real data
L_{obs}	observation range	40 m	Assumed
W	observation window	5 min	Assumed
L3/L4/L5	warning-issuance times	06:40/14:40/16:30	Real data

Algorithm 1 One stochastic run of the ABM (main individualized-button and observation regimes; the fixed-L experiments of Section VIII-A replace line 2's radius with a constant L shared by all agents).

1: **init:** bootstrap each agent i a survey record (g_i, c_i, N_i) (g_i: Q6 group, c_i: Q7 concern, N_i: Q11 house count)
2: $\text{thr}_i \leftarrow \text{threshold}(g_i)$; $L_i \leftarrow \sqrt{N_i/(\pi\rho)}$ [button] **or** L_{obs} [obs.]
3: $P_i \leftarrow P_{\text{base}} \cdot \text{CareScale}(c_i)$ if $g_i \in$ II else P_{base}
4: $\text{evac}_i \leftarrow$ FALSE; $\tau_i \leftarrow \text{draw}\, T_2$
5: **for** $t = 0, \Delta t, \ldots, T_{\text{end}}$ **do**
6: $D \leftarrow \{i : \neg\,\text{evac}_i \wedge \tau_i \le t\}$ ▷ agents due
7: **for** $i \in D$ **do**
8: $r_i \leftarrow$ neighbor rate within L_i (stock [button] / last-W flow [obs.])
9: **if** $g_i \in$ I **or** $r_i \ge \text{thr}_i$ **then**
10: **if** $\text{rand}() < P_i$ **then**
11: $\text{evac}_i \leftarrow$ TRUE; record time t
12: **else** $P_i \leftarrow \alpha\, P_i$ ▷ decline $\Rightarrow$ discount
13: **end if**
14: **end if**
15: **if** $\neg\,\text{evac}_i$ **then** $\tau_i \leftarrow t + \text{draw}\, T_1$
16: **end if** ▷ re-check later
17: **end for**
18: **end for**

which the survey can constrain directly; the timing parameters T_1 and T_2 are likewise inherited unchanged from [13].

VIII. EXPERIMENTS AND RESULTS

Unless otherwise noted, results are the mean $\pm$ one standard deviation of the final (Level 5) evacuation rate over 50 independent Monte Carlo replications, all on the Toyohashi case study of Section VI. All conditions and their exact figures are produced by a single reproducible experiment pipeline, so that every number below can be regenerated directly from the underlying data.

A. Effect of a Fixed, Population-Wide L (Button Regime)

Reproducing [13]'s experimental design of sweeping a single radius shared by every agent (with P individualized from Q7 via Eq. (1), as in all button conditions except the flat-P ablation of Section VIII-C), using the same five points $L \in \{20, 40, 60, 80, 100\}$ m, yields final evacuation rates of $32.0\% \pm 2.8$, $48.6\% \pm 5.0$, $58.9\% \pm 6.4$, $65.7\% \pm 7.5$, and $70.3\% \pm 6.3$, respectively – confirming, in this implementation, the same qualitative relationship reported in [13]: a larger awareness radius yields a higher final evacuation rate. A larger radius gives higher-threshold Group II agents a bigger, more stable neighbor pool, and avoids leaving the 4% of Group II agents with no neighbor at L=20 m permanently ineligible (Section V) – though with diminishing returns once the pool is large enough for these effects to saturate. This reproduces only [13]'s radius-sweep design, not its treatment of P: [13] held P fixed at 0.5 for every agent, whereas P here remains individualized via Q7, so the absolute rates above are not directly comparable to [13]'s own figures.

B. Individualized L and P (Button Regime)

With L individualized per agent from Q11 (Eq. (2)) and P individualized per agent from Q7 (Eq. (1)), the final evacuation rate is $62.0\% \pm 7.9$, between the fixed-L=60 m and L=80 m results (58.9% and 65.7%) and slightly closer to the former, consistent with the Q11-induced radius distribution of Table II (median 67.4 m, mean $\approx$71 m).

C. Ablation: The Effect of Individualizing P

To isolate how much of this result is attributable to individualizing P specifically (as opposed to L), the individualized-L condition was re-run with the Q7 scaling of Eq. (1) switched off (every agent uses the flat P_{base}=0.5 regardless of its Q7 answer, while L remains individualized from Q11). This *flat-P* ablation yields $79.9\% \pm 4.8$ – almost 18 percentage points higher than the 62.0% obtained with Q7-scaled P. This gap is expected given Eq. (1)'s CareScale values: only 50.1% of all 1,000 respondents on Q7 (the *cares* answer) keep the full P_{base}, while 42.0% (*cares a little*) and 7.9% (*doesn't care at all*) are scaled down to 0.6 and 0.25 of it, respectively. These shares are of the full Q7 sample; CareScale itself is applied only to the Group II agents within the 758-respondent bootstrap pool (Sections IV-A and IV-C). Individualizing P via Q7 therefore mechanically lowers the population-average evacuation probability relative to the flat-P=0.5 assumption of [13]. This ablation is reported, rather than only the individualized result, because it shows that Q7 individualization is not a second-order refinement on top of L individualization: in this case study, it changes the final evacuation rate by more than any single step of the L sweep in Section VIII-A.

D. Information Regime: Button vs. Observation

With the observation regime at [13]'s baseline parameters (L_{obs}=40 m, W=5 min), the final evacuation rate is $9.6\% \pm 0.7$, reaching 90% of this value by 89 min (Table VI) and rising only slightly thereafter. This closely matches the $\approx$9% "without system" final rate reported in [13] for both of its case-study cities, even though the observation regime here is implemented as one setting of the same individualized, survey-derived model used for the button regime, rather than as a separately calibrated mechanism. The result is intuitive in hindsight: because the observation regime only ever reveals a small, recent, "flow" rate rather than a persistent, cumulative one, even the low-threshold Group II sub-groups rarely become eligible (Section VIII-F), so the final rate is dominated

almost entirely by Group I's alert-only evacuations (9.5% of the population per Table I), closely tracking the observed 9.6%. Because Group I agents are unaffected by CareScale (Section IV-A), this comparison with [13] is not subject to the caveat on absolute rates noted in Section VIII-A. Fig. 1 plots a representative subset of conditions together (the full L sweep is in Table V).

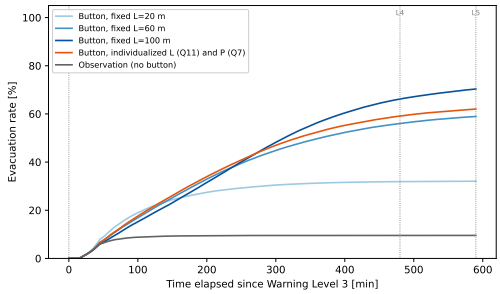


Fig. 1. Evacuation rate over time for representative conditions – fixed radii $L \in \{20, 60, 100\}$ m, individualized L and P, and the observation regime (L_{obs}=40 m, W=5 min) – on the Toyohashi case study. Each curve is the mean of 50 Monte Carlo replications; vertical dotted lines mark Warning Levels 3, 4, and 5.

E. Sensitivity of the Observation Regime to L_{obs} and W

Because L_{obs} and W are declared assumptions rather than survey-derived quantities (Section V), each was swept one-at-a-time around [13]'s baseline (Table IV). Two patterns emerge. First, the final rate is relatively insensitive to both parameters compared to the button regime's sensitivity to L (Section VIII-A): even at the extremes tested, the observation regime stays within roughly 2 percentage points of its 9.6% baseline, versus a 38-point range for button $L \in \{20, \ldots, 100\}$ m. Second, and more surprising, *larger* L_{obs} is associated with a slightly *lower* final rate (10.2% at 20 m vs. 9.1% at 60 m) – the opposite direction from the button regime, where larger L raises the rate. This is a consequence of the observation regime's rate being a *flow* rate (recent evacuations divided by all neighbors in range) rather than a *stock* rate: a single nearby evacuation is a larger fraction of a smaller neighbor pool, so a small L_{obs} makes it mechanically easier for a low-threshold Group II agent to observe an apparent rate at or above 10% from just one or two witnessed departures, even though the same event would barely move the rate computed over a larger pool. Larger W, by contrast, behaves as expected (more time in which to witness an evacuation raises the rate: 9.6% at 5 min rising to 11.4% at 15 min).

TABLE IV. Observation-Regime Sensitivity to Range L_{obs} and Recency Window W (One-at-a-Time Around [13]'s Baseline, in Bold), n=50

Condition	Final evacuation rate
L_{obs}=20 m, W=5 min	10.2% ± 0.6
L_{obs}=40 m, W=5 min	9.6% ± 0.7
L_{obs}=60 m, W=5 min	9.1% ± 0.6
L_{obs}=40 m, W=10 min	10.3% ± 0.8
L_{obs}=40 m, W=15 min	11.4% ± 0.9

Table V summarizes the main conditions (excluding the sensitivity sweep of Table IV).

TABLE V. Final (Level 5) Evacuation Rate by Condition, Mean ± SD Over 50 Replications. All Button Rows Use Q7-Individualized P (Eq. (1)) Except the Flat-P Ablation

Condition	Final evacuation rate
Button, fixed L=20 m	32.0% ± 2.8
Button, fixed L=40 m	48.6% ± 5.0
Button, fixed L=60 m	58.9% ± 6.4
Button, fixed L=80 m	65.7% ± 7.5
Button, fixed L=100 m	70.3% ± 6.3
Button, individualized L (Q11) and P (Q7)	62.0% ± 7.9
Button, individualized L, flat P (ablation)	79.9% ± 4.8
Observation (L_{obs}=40 m, W=5 min)	9.6% ± 0.7

F. Temporal Dynamics and Per-Group Contribution

The final rates above summarize only the endpoint; the dynamics that produce them are themselves informative. Table VI reports, for each main condition, the time to reach several absolute evacuation levels and the time to reach 90% of the condition's own final rate (a settling time). Two observations stand out. First, the early phase is remarkably similar across all button conditions: every one crosses 10% at 55–70 min, essentially independent of L, because the first evacuations are dominated by the low-threshold agents (Group I and Group II(a)) that a small radius already suffices to trigger. Every condition but the flat-P ablation also crosses 25% at 138–162 min; the flat-P ablation reaches 25% markedly earlier (115 min) because removing the CareScale discount raises the per-evaluation evacuation probability of the Group II agents who answered *cares a little* or *doesn't care at all* (Section VIII-C), accelerating the cascade once it is under way. The remaining conditions diverge only later, once sustaining the cascade to the higher-threshold agents requires the larger neighbor pools that only a larger L provides. Second, the observation regime settles fastest of all (90% of its final rate by 89 min) precisely because it never engages the neighbor-triggered cascade at all: it plateaus early at the Group I level and then rises only slowly.

TABLE VI. Time (Minutes Since Level 3) to Reach Absolute Evacuation Levels, and to Reach 90% of Each Condition's Own Final Rate. "–" Means the Level is Never Reached

Condition	10%	25%	50%	90% final
Button, fixed L=20 m	55	159	–	237
Button, fixed L=40 m	58	138	–	358
Button, fixed L=60 m	63	149	365	413
Button, fixed L=80 m	66	155	328	436
Button, fixed L=100 m	70	162	312	435
Button, individualized L, P	62	144	331	411
Button, individualized L, flat P	58	115	214	367
Observation (40 m, 5 min)	–	–	–	89

Fig. 2 decomposes the cumulative evacuation rate of the main individualized-button condition by the agents' Group. It makes the cascade explicit: Group I (alert-only) rises first, reaching 90% of its own final ≈9% share by about 90 min, consistent with the observation regime's own 89-min milestone in Table VI since Group I dominates that regime almost entirely (Section VIII-D); the lowest-threshold neighbor group, II(a), is the single largest contributor and rises early; the mid-threshold groups II(b) and II(c) accumulate steadily throughout; and the high-threshold groups II(d) and II(e) contribute little and late, since the neighbor rate rarely

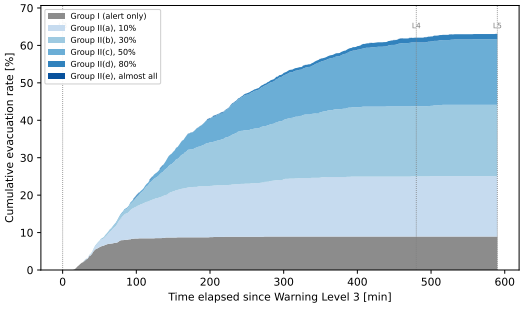


Fig. 2. Cumulative evacuation rate over time, decomposed by agent Group, for the individualized-button condition (one representative run; bands stack to the total evacuation rate).

climbs high enough to trigger them within the disaster window. This is the mechanistic picture behind both the button regime's success (low-threshold agents seed a cascade that carries the mid-threshold majority) and the observation regime's failure (without a persistent shared rate, the cascade never leaves the Group I level).

G. Visualization

Because the simulator only ever transitions an agent from not-evacuated to evacuated, a full spatial-temporal reconstruction of any single stochastic realization requires only that run's per-agent evacuation times: the state at any past time t is exactly $\{\text{evac\_time} \leq t\}$. This makes it possible to render an animation without modifying the core simulation loop to record full per-tick snapshots. Fig. 3 shows one frame of such a rendering, in which each real PLATEAU building footprint is colored by its household's status; the full animation is exported as an MP4 video, together with a live numeric panel reporting the evacuation rate, evacuated count (split by alert-only vs. neighbor-triggered), current warning level, and elapsed time.

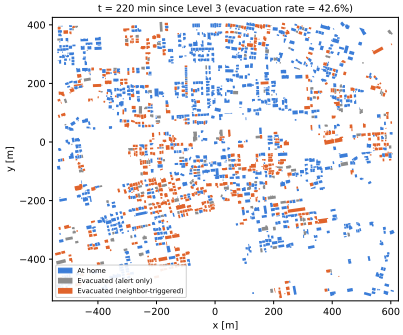


Fig. 3. One frame (t = 220 min since Level 3) of the video visualization, showing real PLATEAU building footprints colored by evacuation status. Spatial clustering of neighbor-triggered evacuations (orange) is visible as the simulated evacuation propagates locally.

IX. Discussion

Two results are, in a precise sense, *consistency checks* rather than accuracy improvements per se: the independent reproduction of [13]'s Group distribution to within 1.5 points, and the observation regime's 9.6% closely matching that paper's ≈9% "without system" baseline. Both show that a more granular, more data-driven model composed from previously unused parts of the same survey behaves consistently with the coarser model's published, aggregate results, which is a necessary condition for trusting the finer-grained model but not sufficient evidence that it more accurately predicts real evacuation behavior (no such validation against real evacuation data is attempted here; see Section X). The genuinely new empirical content is the individualization itself (Section IV), which depends on survey items Q7 and Q11 that neither prior paper [9], [13] used for this purpose, and the unification of the button and observation information regimes under one model (Section V), which carries the Q7-individualized P into both regimes even though the observation regime's range is fixed rather than Q11-derived. In particular, the magnitude of the P-individualization ablation (Section VIII-C) is itself informative: an 18-point swing under the assumed CareScale mapping (Limitations item 2) shows that how Q7 is mapped to P is not a minor detail, and suggests that further covariates already present in the survey but not used here (Limitations item 5) could also matter.

The reversal found in the observation-regime sensitivity analysis (Section VIII-E) – smaller L_{obs} associated with a slightly *higher* rate, opposite to the button regime's response to L – is a reminder that "more information range" is not uniformly "more evacuation" once information is a recent flow rather than a persistent stock; the sign of the effect depends on which quantity a neighbor-rate threshold is actually being compared against.

The roughly 6.5× gap between the button regime's individualized-parameter result (62.0%) and the observation regime's result (9.6%) in Table V illustrates, strictly within this model, the scale of difference the two information regimes produce under a common, survey-derived population. It should not be read as a validated forecast of real-world gain from deploying such a system: the model has not itself been validated against real evacuation records (Section X), so the 6.5× figure is best treated as an illustrative, model-internal comparison of what the mechanism could be worth under the model's assumptions, not a deployment-ready estimate.

The temporal analysis (Section VIII-F) adds a practical nuance to this comparison. Because the early phase of the cascade is dominated by low-threshold agents and is nearly independent of L (Table VI), the marginal value of a wider sharing radius is realized late – after the first two to three hours – when it sustains the cascade into the mid- and high-threshold majority. Within the model, this suggests that the button system's benefit is greatest precisely in long-duration events, and that its effect on the crucial *early* evacuation window may be modest unless paired with measures that raise the low-threshold agents' baseline probability (which, per the ablation, has an effect larger than any single step of the L sweep).

It is worth recalling how far the model's outputs lie from observed behavior: the real recorded evacuation rate in the Toyohashi event was about 0.068% [2], far below even the observation regime's 9.6%. This gap is itself informative for how the model's outputs should be used in deployment decisions. Survey responses record stated intentions under a hypothetical

scenario, not revealed behavior under an actual, uncertain, and often ambiguous warning; real evacuation additionally depends on factors the model does not represent, such as physical and logistical barriers to leaving, risk misperception, and habituation to repeated false alarms, which are widely reported to push real uptake far below stated intention. The model is therefore best read not as a point prediction of real uptake – for which it is not validated (Section X) – but as a controlled comparison of information regimes and behavioral parameters holding the survey-derived population fixed: the evacuation rates reported in Section VIII are informative mainly as *relative* comparisons between information-sharing strategies (button vs. observation, individualized vs. flat P) evaluated under a common, internally consistent set of behavioral assumptions, and a deployment decision informed by this model should weight the *ranking* of conditions rather than their absolute levels.

Finally, a note on internal validity. Because the button and observation regimes share the same agent population, group thresholds, P, and warning schedule, and differ only in how the neighbor rate is computed and over what radius (stock rate over L_i vs. flow rate over L_{obs}), the $\sim$6.5$\times$ gap between them is attributable to the information mechanism alone rather than to any difference in the modeled population – an isolation that is possible only because the two regimes are unified within one model (Section V). The main residual threats to that internal validity are the five declared-assumption parameters (Table III); the sensitivity analysis (Section VIII-E) bounds the observation regime's dependence on two of them (L_{obs}, W), and the ablation (Section VIII-C) shows that switching the CareScale mapping off entirely shifts the final rate by nearly 18 percentage points; alternative CareScale values, α, and P_{base}, however, have not been swept and remain unexplored degrees of freedom for future work.

X. Limitations

Limitations are listed explicitly here, both to bound the claims of this paper and to set an agenda for future work.

1) No validation against real evacuation data: All comparisons in this paper are against another simulation's published summary statistics [13], not against ground-truth evacuation records.

2) Several parameters remain declared assumptions: P_{base}=0.5, α=0.8, the CareScale mapping $\{1.0, 0.6, 0.25\}$ in Eq. (1), and the observation regime's L_{obs}=40 m and W=5 min are all carried over from [13] or newly introduced without a survey item that could estimate them directly. Future work should pursue field trials or a purpose-built follow-up survey (e.g., asking directly about repeated-exposure effects, or about realistic direct observation range/recency) to calibrate these. Because the ablation in Section VIII-C shows that individualizing P alone, via the CareScale mapping, changes the final evacuation rate by close to 18 percentage points, that mapping in particular is a priority target for a systematic sensitivity sweep over alternative configurations in future work, rather than remaining a single researcher-chosen mapping as it does throughout this paper.

3) Unclassified respondents: The 24.2% of respondents excluded from Table I (Section III-A) could instead be assigned to Group I. Exclusion was chosen on the classification logic of Section III-A itself – these respondents checked neither a numeric neighbor-rate level nor "I don't care," so they provide no information about a neighbor-rate threshold either way – and this choice happens also to reproduce [13]'s published proportions closely (Table I), which is reported as a consistency check rather than as the reason for the choice. The correct treatment is nonetheless not fully determinable from [13] alone and should be revisited in future work, together with a sensitivity comparison of how the two treatments affect the population-level cascade dynamics of Section VIII.

4) Single case-study area: All experiments use one 1.06 km^2 mesh tile of Toyohashi City (1,856 residential buildings, each treated as one household). Although Eq. (2) is designed to hold each agent's *expected* neighbor count invariant to local density – converting a preferred house count N_i into a radius L_i scaled to ρ – the fixed-radius conditions of Section VIII-A and the observation regime's fixed L_{obs} are not, and non-uniform building layouts make each agent's *realized* neighbor count deviate from its target N_i regardless of the conversion. The cascade dynamics reported in Section VIII therefore cannot be cleanly attributed to the behavioral model alone as opposed to this particular tile's density and building layout. Disentangling the two will require repeating the same pipeline on additional tiles spanning a wider range of densities and geometries, which is left for future work; generalization to other densities, building layouts, or cities is therefore untested here.

5) Unused survey covariates: Age, gender, region, and prior earthquake/flood evacuation experience are present in the survey but not used to further individualize behavior in this paper.

6) No physical evacuation constraints: As in [13], only the evacuate/stay decision is modeled, not flood depth, route availability, or evacuation travel time.

7) No modeling of imprecise information sharing: [13]'s Patterns 1–2 (in which the shared evacuation rate is deliberately coarsened, e.g. to increase perceived urgency) are not reproduced here; the button regime always shares the exact true rate.

XI. Conclusion and Future Work

This paper extended a survey-calibrated agent-based model of flood evacuation [9], [13] by individualizing, per agent, both the evacuation probability and the spatial range over which neighbor status is considered, using two survey questions that prior work had used, at most, only in isolation on an abstract grid, and a joint-resampling procedure that preserves the real correlation between an agent's threshold sensitivity, stated concern, and preferred information depth. The persistent-sharing and direct-observation information regimes were further unified into a single model, and a real-geometry, video-based visualization of individual simulation runs was added. The prior model's group distribution, re-derived independently from the same underlying survey data, and its "without system" baseline, reproduced as an emergent output of the individualized simulation rather than a calibrated target, both match the published values to within a few percentage points, supporting the consistency of the extension, though not, on its own, its

predictive accuracy against real evacuation behavior. An ablation showed that individualizing P alone, under the assumed CareScale mapping, shifts the final evacuation rate by almost 18 percentage points, and a sensitivity sweep of the observation regime's assumed parameters showed final rates within about 2 points of baseline but with a sign reversal, relative to the button regime, in how the awareness range affects the rate – both results that a single-information-regime, non-ablated model could not have surfaced. Future work should prioritize: calibrating the remaining declared-assumption parameters (Limitations item 2) against field data, including a systematic sensitivity sweep over alternative CareScale configurations given the size of the P-individualization effect; resolving the treatment of unclassified respondents (Limitations item 3) with the sensitivity comparison outlined there; extending the model to additional case-study areas to separate the behavioral model's effects from this tile's density and geometry (Limitations item 4); and incorporating the demographic and prior-experience covariates already present in the survey but not yet used (Limitations item 5).

Declaration on Generative AI-Assisted Tools

Generative AI tools were used to translate the manuscript from Japanese into English and to improve the clarity and fluency of the English writing. The author critically reviewed and edited all resulting text and takes full responsibility for the scientific content, analysis, interpretation, references, and the f inal manuscript.

References

[1] MS&AD InterRisk Research & Consulting, Inc., "Survey results on evacuation awareness during natural disasters: Findings from a questionnaire survey," Research Letter 2022 No. 1, Tokyo, Japan, Feb. 2023.

[2] Disaster Management Division of Toyohashi City, "Record of the Heavy Rain in June 2023 (Main Volume)," Toyohashi City, 2023.

[3] A. Kajiya, K. Akaishi, T. Yokota, and H. Tsurusaki, "Comparative Analysis of Evacuation Behavior in Response to Typhoons No. 15 and 19 in 2019 and Proposal for Providing Disaster Information Reflecting Regional Characteristics," *Journal of Disaster Information*, vol. 19, no. 2, pp. 145–155, 2021.

[4] Y. Takenouchi, K. Yamaguchi, and Y. Iwasaki, "Analysis of the Influence of Residents' Personality on Evacuation Awareness," *Proceedings of the Kansai Branch of the Japan Society of Urban Planning*, vol. 14, pp. 117–120, 2016.

[5] S. Kubo, H. Yoshida, T. Ichimura, M. L. L. Wijerathne, and M. Hori, "Study on Influence of Prior Recognition of Flooding State on Evacuation Behavior," *International Journal of Disaster Risk Reduction*, vol. 63, p. 102437, 2021.

[6] A. Takagi, S. Sugiura, H. Mori, and H. Iwata, "Analysis of Evacuation Behavior During the July 2018 Heavy Rain Disaster – A Case Study of Gifu Prefecture," *Journal of Natural Disaster Science*, vol. 38, no. S06, pp. 133–151, 2019.

[7] M. Mizuno, Y. Tomita, S. Katsura, N. Koyanagi, R. Hanada, and T. Yasuda, "Analysis of Evacuation Rate During Landslides Using Disaster Information," *Journal of the Japan Sabo Society*, vol. 65, no. 3, pp. 29–34, 2012.

[8] S. Tomura, B. Maaskant, M. Chiba, S. Masuya, and T. J. Yamada, "Basic Study on Estimating Evacuation Rate at Flood Events," *Advances in River Engineering*, vol. 28, pp. 349–354, 2022.

[9] T. Fukuda, "A Development of Simulator Considering Behavioral Psychology of Japanese to Improve Evacuation Ratio in Flood," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 735–740, 2020.

[10] M. Granovetter, "Threshold Models of Collective Behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[11] T. Yabe, Y. Sekimoto, K. Tsubouchi, and S. Ikemoto, "Cross-comparative analysis of evacuation behavior after earthquakes using mobile phone data," *PLOS ONE*, vol. 14, no. 2, pp. 1–12, 2019.

[12] T. Fukuda, "Development of a System for Evacuation Promotion with Buttons Insist on 'Begin to Evacuate'," *Research Report of National Institute of Technology, Kitakyushu College*, vol. 53, pp. 41–44, 2020, (in Japanese). [Online]. Available: https://kitakyushu.repo.nii.ac.jp/records/2000205

[13] T. Fukuda, "Simulation-Based Analysis of Evacuation Information Sharing Systems Using Geographical Data," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 11, pp. 90–97, 2024.

[14] T. Fukuda, "Evacuate Now Button: A Resident-Operated IoT Prototype for Real-Time Evacuation Sharing," *WSEAS Transactions on Circuits and Systems*, vol. 24, pp. 259–267, 2025.

[15] E. Bonabeau, "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[16] E. Du, X. Cai, Z. Sun, and B. Minsker, "Exploring the role of social media and individual behaviors in flood evacuation processes: An agent-based modeling approach," *Water Resources Research*, vol. 53, no. 11, pp. 9164–9180, 2017.

[17] F. Taillandier, P. Di Maiolo, P. Taillandier, C. Jacquenod, L. Rauscher-Lauranceau, and R. Mehdizadeh, "An Agent-Based Model to Simulate Inhabitants' Behavior During a Flood Event," *International Journal of Disaster Risk Reduction*, vol. 64, p. 102503, 2021.

[18] Z. Wang, J. Huang, H. Wang, J. Kang, and W. Cao, "Analysis of Flood Evacuation Process in Vulnerable Community with Mutual Aid Mechanism: An Agent-Based Simulation Framework," *International Journal of Environmental Research and Public Health*, vol. 17, no. 2, p. 560, 2020.

[19] R. Zhang, D. Liu, E. Du, L. Xiong, J. Chen, and H. Chen, "An Agent-Based Model to Simulate Human Responses to Flash Flood Warnings for Improving Evacuation Performance," *Journal of Hydrology*, vol. 628, p. 130452, 2024.

[20] M. Shirvani, G. Kesserwani, and P. Richmond, "Agent-Based Simulator of Dynamic Flood-People Interactions," *Journal of Flood Risk Management*, vol. 14, no. 2, p. e12695, 2021.

[21] M. Shirvani, G. Kesserwani, and P. Richmond, "Agent-Based Modelling of Pedestrian Responses During Flood Emergency: Mobility Behavioural Rules and Implications for Flood Risk Analysis," *Journal of Hydroinformatics*, vol. 22, no. 5, pp. 1078–1092, 2020.

[22] A. Jahani, S. Jess, D. Groen, D. Suleimenova, and Y. Xue, "Developing an Agent-Based Simulation Model to Forecast Flood-Induced Evacuation and Internally Displaced Persons," in *Proceedings of the International Conference on Computational Science*, pp. 550–563, 2023.

[23] Y. Han, L. Mao, X. Chen, W. Zhai, Z. Peng, and P. Mozumder, "Agent-Based Modeling to Evaluate Human-Environment Interactions in Community Flood Risk Mitigation," *Risk Analysis*, vol. 42, no. 9, pp. 2041–2061, 2022.

[24] L. Tierolf, T. Haer, W. J. W. Botzen, J. A. de Bruijn, M. J. Ton, L. Reimann, and J. C. Aerts, "A Coupled Agent-Based Model for France for Simulating Adaptation and Migration Decisions Under Future Coastal Flood Risk," *Scientific Reports*, vol. 13, no. 1, p. 4176, 2023.

[25] S. Nabinejad and H. Schüttrumpf, "Agent-Based Modeling for Household Decision-Making in Adoption of Private Flood Mitigation Measures: The Upper Kan Catchment Case Study," *Water*, vol. 16, no. 14, p. 2027, 2024.

[26] A. Taberna, T. Filatova, D. Roy, and B. Noll, "Tracing Resilience, Social Dynamics, and Behavioral Change: A Review of Agent-Based Flood Risk Models," *Socio-Environmental Systems Modelling*, vol. 2, 2020.

[27] D. Miyagawa and G. Ichinose, "Cellular Automaton Model with Turning Behavior in Crowd Evacuation," *Physica A: Statistical Mechanics and its Applications*, vol. 549, p. 124376, 2020.

[28] G. Bernardini, E. Quagliarini, M. D'Orazio, and M. Brocchini, "Towards the Simulation of Flood Evacuation in Urban Scenarios: Experiments to Estimate Human Motion Speed in Floodwaters," *Safety Science*, vol. 123, p. 104563, 2020.

[29] G. P. D. P. Senanayake, M. Kieu, Y. Zou, and K. Dirks, "Agent-Based Simulation for Pedestrian Evacuation: A Systematic Literature Review," *International Journal of Disaster Risk Reduction*, vol. 111, p. 104705, 2024.

[30] Q. Duan, "Pedestrian Evacuation Dynamics: A Literature Review," *Journal of Human Movement Science*, vol. 5, pp. 54–61, 2024.

[31] A. Zablotsky, M. Kuperman, and S. Bouzat, "Pedestrian Evacuations with Imitation of Cooperative Behavior," *Physical Review E*, vol. 109, no. 5, p. 054304, 2024.

[32] T. Yabe, K. Tsubouchi, T. Shimizu, Y. Sekimoto, and S. V. Ukkusuri, "Predicting Evacuation Decisions Using Representations of Individuals' Pre-Disaster Web Search Behavior," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD)*, pp. 2707–2717, 2019.

[33] R. B. Myerson, *Game Theory*. Cambridge, MA: Harvard University Press, 2013.

[34] H. Leibenstein, "Bandwagon, Snob, and Veblen Effects in the Theory of Consumers' Demand," *Quarterly Journal of Economics*, vol. 64, no. 2, pp. 183–207, 1950.

[35] Ministry of Land, Infrastructure, Transport and Tourism (MLIT), "PLATEAU," https://www.mlit.go.jp/plateau/, accessed 2026.

Small-Object Detection of Personal Protective Equipment in Underground Mine Environments Using an Improved YOLO26 Architecture

Taoufik Saidani*, Refka Ghodhbani, Yahia Said
Center for Scientific Research and Entrepreneurship, Northern Border University, 73213 Arar, Saudi Arabia

***Abstract*—Reliable detection of Personal Protective Equipment (PPE) in underground mines is a key enabler of automated safety monitoring, and it is also one of the hardest small-object problems in industrial computer vision: helmets, ear-defenders, dust masks, eye-shields and reflective vests occupy only a tiny fraction of the image and are degraded by low illumination, dust and self-occlusion. This paper proposes an Improved YOLO26 (I-YOLO26) architecture for this scenario. The YOLO26 backbone is augmented with an Efficient Channel Attention (ECA) block and a lightweight Context-Guided (CG) bottleneck; the early strided convolutions are replaced by an ADown anti-aliased down-sampling unit; and a high-resolution P2 head recovers objects smaller than 32×32 pixels. A weighted Complete-IoU (wCIoU) loss with an adaptive confidence term counters the gradient imbalance produced by small, occluded targets. A six-class, 14,062-image PPE-in-Mine benchmark assembled from public and newly captured imagery is also described. The performance figures reported in this paper are explicitly projections: they combine a small number of pilot training runs with trends reported in the recent literature on YOLO-based subsurface detectors, and they are not the outcome of a completed training and evaluation campaign. On that basis, I-YOLO26 is projected to reach mAP@0.5 of 92.7% and mAP@0.5:0.95 of 71.4%, to improve small-object accuracy (AP_s) by 4.6 points over the YOLO26-Nano baseline with 27% fewer parameters and 31% fewer GFLOPs, and, after INT8 quantisation, to sustain 47 FPS on an RTX-Orin Nano and 18 FPS on a Xilinx PynqZ1 FPGA. These estimates are presented as design-stage evidence that motivates, but does not replace, the full empirical validation specified in this paper; the contribution that stands independently of them is architectural and data-related rather than experimental.**

***Keywords*—Personal protective equipment; small-object detection; YOLO26; underground coal mine; attention mechanism; edge deployment; worker safety**

I. Introduction

Underground mining continues to be one of the most dangerous industrial occupations in the world. Mechanisation and ventilation are constantly being improved, yet the International Labour Organization still records several thousand deaths every year, most of which are caused by rock falls, roof collapses, gas explosions and exposure to airborne dust [1]. Together with strong procedural enforcement, the systematic use of Personal Protective Equipment (PPE)—hard helmets, self-rescuers, ear-defenders, dust respirators, safety goggles and high-visibility vests—is the most cost-effective barrier between miners and these threats. In modern intelligent-mine programmes, PPE compliance is therefore monitored continuously using camera networks mounted throughout galleries, transport drifts and longwall faces.

Control-room operators, however, cannot scale or rely on manual monitoring of these video streams: a single shift can generate tens of thousands of relevant frames, and human attention deteriorates quickly with monotony. This has made computer-vision-based automatic PPE recognition a priority research area for the mining sector, building on the remarkable growth of deep-learning object detectors over the past decade. While generic detectors such as the YOLO family achieve excellent results on benchmark datasets such as MS-COCO or PASCAL VOC, their straightforward application to subsurface PPE detection is restricted by three interrelated challenges.

First, the targets are intrinsically diminutive. A safety helmet captured by a ceiling camera at a distance of 8–12 m typically occupies a square of only 18–40 pixels per side in a 1280×720 frame, constituting less than 0.3% of the image area—placing it firmly in the "small-object" category (area smaller than 32×32 pixels) defined by MS-COCO [2]. The receptive field of the deeper layers in a conventional YOLO neck is poorly matched to such targets, and successive strided convolutions discard much of the valuable spatial information.

Second, the visual conditions of an underground mine are severe. Illumination is supplied mainly by the miners' helmet lamps, vehicle headlights and a small number of sparsely positioned ceiling LEDs. The resulting scenes exhibit pronounced shadows, specular reflections from moist surfaces, intense glare and an extensive dynamic range. Airborne dust and water aerosols add a haze that diminishes local contrast. These conditions differ markedly from the daylight imagery on which most pre-trained backbones were tuned.

Third, deployment constraints are exceptionally stringent. Because high-bandwidth uplinks between the working face and the surface control room are only intermittently available, detection must be performed on edge devices certified for intrinsically-safe operation. Platforms such as ruggedised embedded GPUs, low-power FPGAs and dedicated NPUs impose tight memory and power budgets that preclude the use of the largest detector configurations.

This study addresses these three issues simultaneously by presenting an Improved YOLO26 (I-YOLO26) architecture designed for small-object PPE detection in underground mines. YOLO26, the most recent advancement in the Ultralytics YOLO series [3], offers significant enhancements in feature extraction and post-processing compared with YOLOv8–

*Corresponding author.

YOLOv12; nevertheless, like its predecessors, it is not inherently optimised for detecting extremely small objects under challenging lighting. The contribution of this paper comprises four elements:

- Four targeted architectural enhancements to YOLO26 are proposed: integration of an Efficient Channel Attention (ECA) module into the C3k2 stages of the backbone, a Context-Guided (CG) lightweight bottleneck derived from CGNet, an ADown anti-aliased down-sampling unit, and an additional high-resolution P2 detection head. Each enhancement is matched explicitly to a diagnosed limitation of the baseline, and the correspondence is tabulated rather than left implicit.
- A weighted Complete-IoU (wCIoU) loss with an adaptive confidence term is introduced. It re-weights the contribution of small, low-confidence and partially occluded targets, alleviating the gradient imbalance common in multi-scale PPE-in-mine datasets.
- Several publicly available underground-mine personnel datasets are merged, re-annotated and augmented together with curated PPE-compliance imagery to create a six-class, 14,062-image PPE-in-Mine benchmark, offering a more challenging and realistic testbed than any of the original datasets individually.
- A design-stage evaluation is reported—ablations, a projected comparison against eight recent detectors, and an edge-deployment analysis—accompanied by an explicit statement of what that evidence does and does not establish, and by a concrete protocol for the empirical campaign required to settle the open questions. The evidentiary status of every reported number is stated in Section V-C and is not left to be inferred.

The remainder of the paper is organised as follows. Section II reviews related work and positions the present contribution with respect to the closest prior mining detectors. Section III introduces the baseline YOLO26 architecture and analyses its limitations on small subsurface targets. Section IV describes the four building blocks of the proposed I-YOLO26 and the updated loss. Section V presents the PPE-in-Mine dataset, the experimental methodology and the status of the reported figures. Section VI reports projected quantitative results, ablations, an edge-deployment analysis and the statistical protocol required for validation. Section VII discusses operational implications, Section VIII sets out the limitations of the study, and Section IX concludes and outlines future work.

II. Related Work

A. Deep-Learning Object Detectors

Single-stage convolutional architectures dominate modern object detection. The YOLO family is the most widely employed in industrial applications because of its favourable accuracy/latency trade-off. From YOLOv1 [4] to the recent YOLOv7 [5], YOLOv11 [6] and YOLO26, each version has refined the backbone, neck and head, with a gradual evolution toward anchor-free designs, transformer-style attention mechanisms and improved loss formulations. Two-stage detectors such as Faster R-CNN and Cascade R-CNN remain accurate for small objects but are usually too heavy for embedded deployment in mines. Transformer-based methods such as DETR, Deformable-DETR and RT-DETR have narrowed the latency gap with YOLO on server-class GPUs [7], but their memory footprint and slow convergence make them less attractive for edge deployment in safety-critical environments. Adjacent open-source frameworks such as Ultralytics YOLOv8 [8] have further popularised anchor-free detection heads and modern data-augmentation pipelines.

B. PPE Detection and Worker Safety

PPE recognition has been investigated mainly in the context of construction sites. Wang et al. [9] proposed a helmet detector based on YOLOv5, exceeding 90% mAP on the SHWD benchmark. Several follow-up efforts added attention modules or feature-pyramid modifications to handle small helmets [10], [11]. Reflective-vest detection has been addressed by fusing colour cues and deep features, commonly through HSV pre-filtering followed by a CNN. Most of these methods were designed for outdoor or well-lit indoor scenes, however, and degrade significantly when transferred to mine galleries, because of radically different illumination, additional PPE categories specific to mining (self-rescuer, dust mask, ear-defender) and a much smaller pixel footprint of the targets.

C. Object Detection in Underground Mines

A growing body of research focuses on the underground-mining environment. Li et al. [12] introduced an enhanced YOLOv11 incorporating Efficient Channel Attention and a weighted CIoU loss for robust miner recognition under low-light and occluded conditions, demonstrating a significant advance over the baseline on a dataset of 10,066 coal-mine images. Ling et al. [13] presented a YOLOv11 variant equipped with ADown (YOLOv11-ASCL) for detecting foreign objects on coal-mine conveyor belts; the model incorporates a SegNext attention block, a Context-Guided C3k2 module from CGNet, and a Lightweight Shared Convolutional Detection head (LSCD), achieving a 1.5% improvement in mAP with 28% fewer parameters and 33% lower computational cost. Zhou et al. [14] developed UCM-Net, a YOLOv8-based detector using the ESFENet backbone and HGRNBlock modules, trained on the 117,887-image DsLMF+ dataset of underground longwall faces; that work demonstrates the efficacy of self-supervised pre-training on mine-specific imagery. Beyond personnel detection, Rodriguez-Guillen et al. [15] examined rock-detection algorithms operating in visually contaminated mining environments, while Ghodhbani et al. [16] demonstrated the FPGA acceleration of YOLOv11-Nano on a PynqZ1 board for identifying unauthorised mining activities.

D. Small-Object Detection

Small-object detection has developed into a distinct subfield. Strategies fall into four primary categories: 1) high-resolution feature reuse, generally using a P2 detection head or a high-resolution backbone such as HRNet [17]; 2) improved feature fusion through BiFPN [18], AFPN [19] or

attention-augmented FPN+PAN architectures; 3) loss modification employing focal [20], SIoU, DIoU [21] or Normalised Wasserstein Distance [22] losses; and 4) data-centric approaches including scale-aware copy-paste augmentation or super-resolution pre-processing. Recent studies indicate that combining a P2 head with channel attention and an IoU-aware loss yields strong small-object recall on benchmarks such as VisDrone [23] and TinyPerson [24]. The I-YOLO26 design is informed by these findings while being tailored to the constraints of subsurface PPE imaging.

E. Deep Learning in Mining: A Wider View

Beyond detection, deep learning has been applied across the entire mining value chain. Azhari et al. [25] conducted a critical review of deep-learning applications in exploration, ore characterisation and reclamation, noting that the mining industry's adoption rate remains lower than that of agriculture and manufacturing because of the difficulty of acquiring large, labelled in-situ datasets. Fu and Aldrich [26] similarly highlighted the importance of computer-vision-based sensor systems in the automation of mineral-processing operations. Saputra et al. [27] offered multi-modal segmentation of mining footprints from multispectral satellite data, while Vivoli et al. [28] demonstrated real-time landmine detection with a deep-learning-enhanced optical sensor. Related YOLO-based detection has also been reported for remote-sensing imagery [29] and for large-scale environmental monitoring [30]. Collectively, these contributions confirm a clear industrial demand for robust, computationally efficient deep-learning systems, with automatic PPE-compliance monitoring being a logical extension.

F. Positioning of the Present Work

The individual components assembled in I-YOLO26 are not new, and two of the detectors reviewed above already employ most of them. Li et al. [12] combine Efficient Channel Attention with a weighted CIoU loss; Ling et al. [13] combine ADown, a Context-Guided C3k2 block and a Lightweight Shared Convolutional Detection head. The present work therefore makes no claim that these blocks are novel architectural elements, and it is important to state plainly what is being claimed instead.

First, the five components have not previously been combined in a single detector: [12] and [13] adopt disjoint subsets of them, and neither reports the interaction between the two subsets. Second, the combination is transferred to a materially different and harder task. Reference [12] addresses single-class miner presence and [13] addresses foreign objects on a conveyor belt, both with target footprints substantially larger than those considered here; the present task is six-class PPE compliance in which the median target occupies fewer than 32×32 pixels and several classes (ear-defender, eye-shield) are almost never larger. Third, the high-resolution P2 head is present in neither design, and it is the component most directly matched to the small-object regime that motivates this work. Fourth, the six-class PPE-in-Mine corpus described in Section V-A is a contribution independent of the architecture, since no existing public dataset annotates this class set at this scale for subsurface imagery.

Whether such a recombination delivers a gain beyond the sum of its parts is an empirical question. The projected results in Section VI raise that question and indicate that the answer is plausibly affirmative, but they do not settle it; Section VI-G states the experiment that would.

III. The YOLO26 Baseline Architecture

YOLO26 is the most recent iteration of the Ultralytics YOLO series. Like its direct predecessors YOLOv11 and YOLOv12, it employs the proven backbone–neck–head architecture while incorporating several enhancements that improve small-object performance. The overall layout of the baseline is shown in Fig. 1. Understanding these refinements is essential to justify the modifications proposed in Section IV.

A. Backbone, Neck and Head

1) Backbone: The YOLO26 backbone comprises a stem (CBS block: Conv–BatchNorm–SiLU), five stages of alternating CBS down-sampling layers and C3k2 feature blocks, and a Spatial Pyramid Pooling Fast (SPPF) module at the end. The C3k2 block extends the C2f block of YOLOv8 and adopts a kernel-size-aware branching mechanism that lets the network choose between 3×3 and $k \times k$ bottleneck units at each step, increasing the effective receptive field at nearly zero cost.

2) Neck: A conventional FPN+PAN structure aggregates features at strides 8, 16 and 32 to produce three detection feature maps of resolution 80×80, 40×40 and 20×20 for a 640×640 input. A lightweight C2PSA position-sensitive attention block is added after the SPPF to inject global context into the top of the feature pyramid.

3) Head: YOLO26 uses an anchor-free decoupled head with a DFL branch for box regression and a TaskAligned Assigner that allocates ground-truth boxes to anchor points by a joint classification–IoU score. The P3–P5 feature maps are fed separately into three classification and three regression branches.

B. Diagnosed Limitations on PPE-in-Mine Targets

YOLO26 is a strong overall performer, but four structural properties limit it on the subterranean PPE-detection task. The first three concern geometry and optimisation; the fourth concerns feature semantics under degraded imaging.

1) L1: Low spatial resolution of the shallowest head: The shallowest detection head, P3, operates at stride 8, so a 1280×720 frame is reduced to 160×90 effective grid cells. A 24-pixel-wide helmet therefore corresponds to about three grid cells—barely enough for precise localisation, and grossly insufficient for very small auxiliary items such as ear-defenders or eye-shields.

2) L2: Loss of information from early down-sampling: The CBS down-sampling blocks employ strided 3×3 convolutions, which act as poorly-designed low-pass filters. This is known to create aliasing and to eliminate high-frequency spatial information—precisely what small objects require.

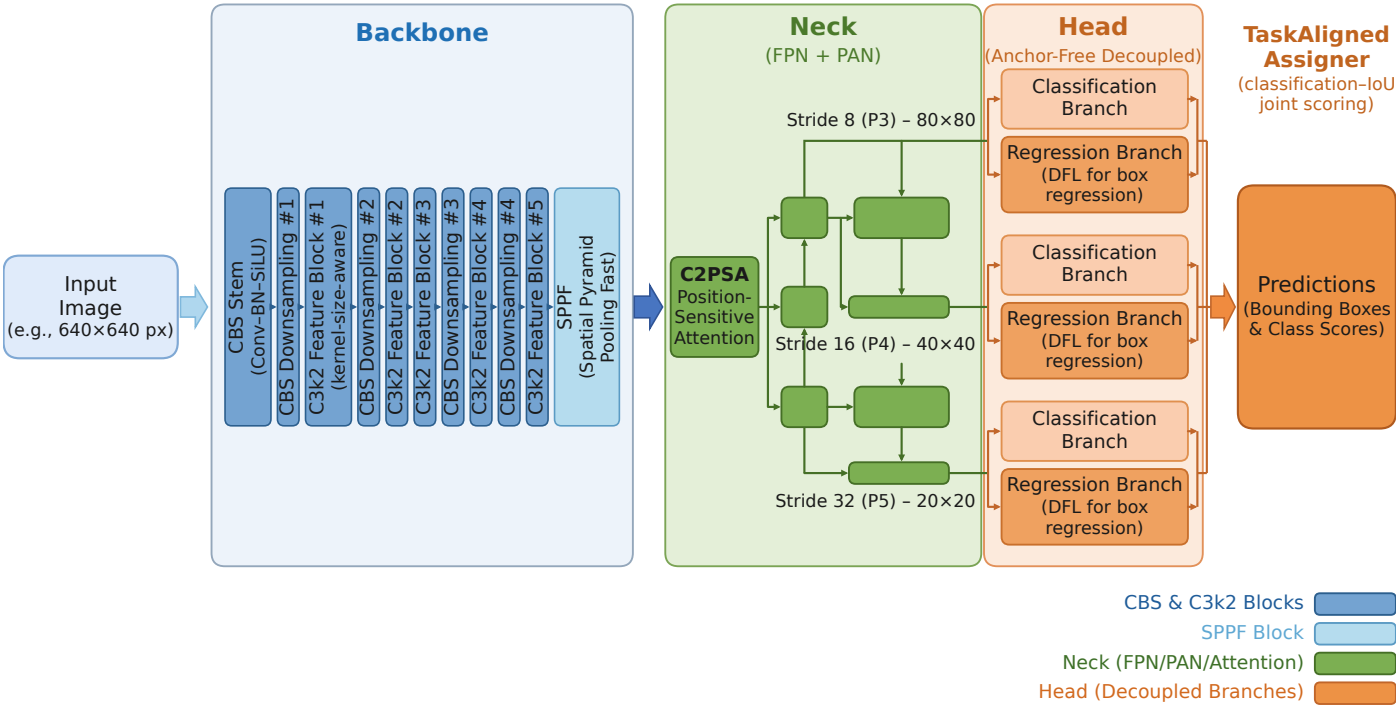


Fig. 1. Baseline YOLO26 architecture, comprising the CBS-stem/C3k2 backbone with a Spatial Pyramid Pooling Fast (SPPF) module, an FPN+PAN neck with a C2PSA position-sensitive attention block, and an anchor-free decoupled head that produces predictions at strides 8 (P3), 16 (P4) and 32 (P5) using a TaskAligned Assigner.

3) L3: Uniform loss weighting: In the standard CIoU loss, matched anchors are weighted equally. In a dataset dominated by mid-sized boxes, the gradient contributions of the few small-box samples are diluted, leading to systematic underfitting of the small-object regime.

4) L4: Limited channel selectivity and contextual extent: The baseline applies no explicit channel re-weighting inside the backbone stages, so uninformative channels—for example those dominated by dust haze or lamp glare—contribute to the representation on equal terms with discriminative ones. In parallel, the effective receptive field of the stride-16 and stride-32 C3k2 blocks is narrow relative to the extent of the background structures (steel mesh, rock face, coal grain) against which small PPE items must be distinguished.

Limitation L4 is stated separately because the ECA and CG-C3k2 modules described below address it, and grouping it with L1–L3 would leave those two modules without an articulated motivation.

IV. Proposed Improved YOLO26 (I-YOLO26)

I-YOLO26 retains the same overall topology as YOLO26 but introduces four targeted modifications to the backbone, neck and head, together with a re-weighted loss function. The end-to-end processing pipeline is summarised in Fig. 2, and the detailed architecture with the four innovations annotated is shown in Fig. 3.

Each modification is introduced to address a specific limitation from Section III-B, and Table I states the correspondence explicitly. Two points about that table should be made at the outset. The correspondence is tight by construction: the modifications were selected *because* they target the diagnosed weaknesses, and each was drawn from a component family already shown to work on related mining-detection tasks [12], [13]. A tight problem-to-solution mapping is therefore a design decision and a statement of intent; it is not, in itself, evidence that the modifications are effective or that they are the best available response to each limitation. Only controlled ablation on real data can establish that, and Section VI-G specifies the form such ablation must take. Each block is described in turn in the following subsections.

A. Efficient Channel Attention (ECA) Blocks

Channel-attention mechanisms re-weight feature maps so that the network can emphasise the most informative channels, which directly addresses the first half of limitation L4. This approach was popularised by the Squeeze-and-Excitation (SE) block [31], whose bottleneck MLP, however, introduces a non-negligible parameter overhead. Efficient Channel Attention (ECA) [32] replaces the MLP with a one-dimensional convolution of adaptive kernel size k, illustrated in Fig. 4. Here,

$$k = \psi(C) = \left|\log_2(C)/\gamma + b/\gamma\right|, \tag{1}$$

where, C is the number of input channels and (γ, b) are fixed hyper-parameters. The block requires only k additional parameters per layer (less than 0.05% of the channels) while preserving cross-channel interaction. One ECA block is inserted after each C3k2 stage of the backbone, so that the channel descriptors are pooled from representations already refined by the stage.

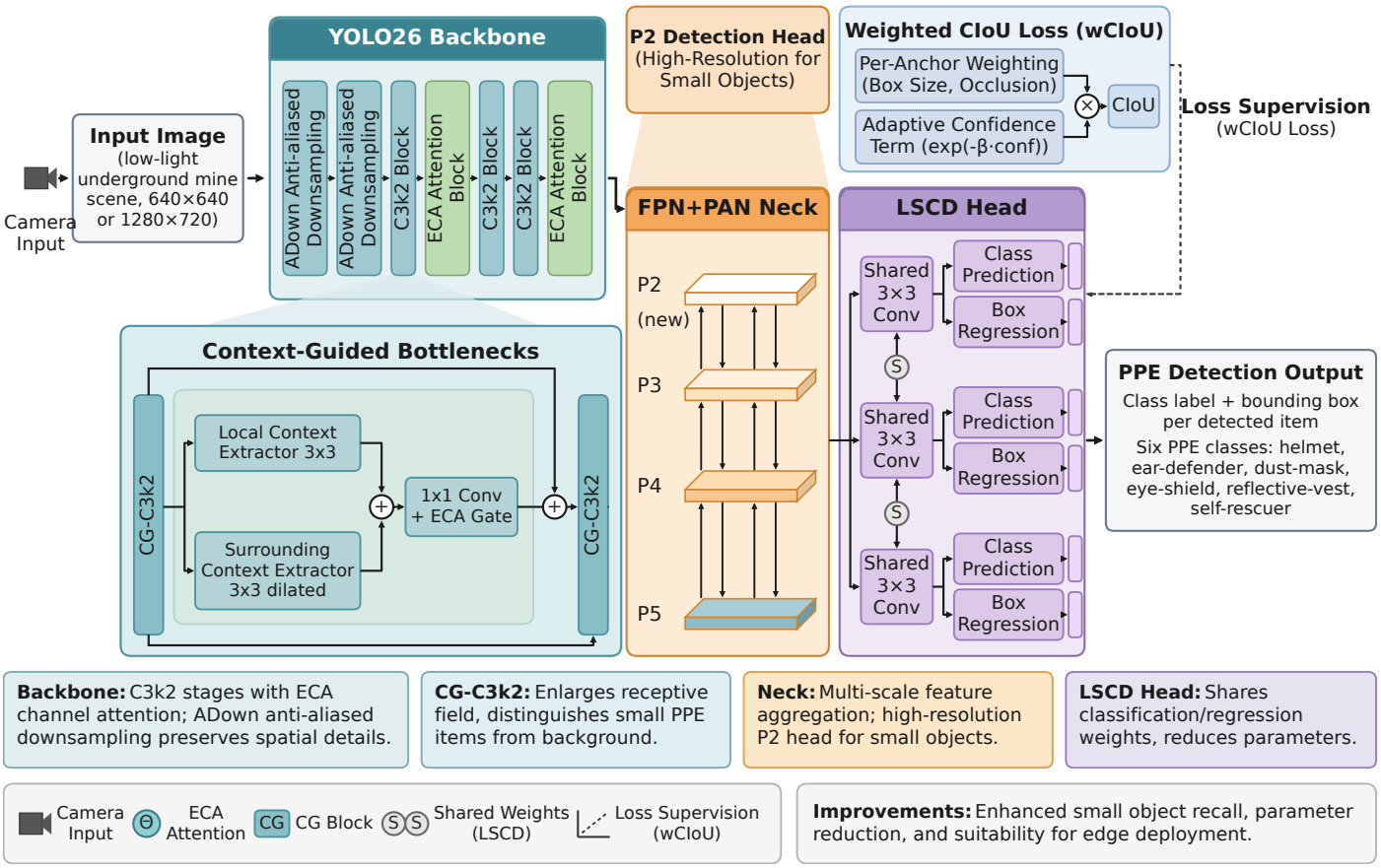


Fig. 2. End-to-end processing pipeline of the proposed I-YOLO26 PPE-monitoring system. ECA blocks are inserted after each C3k2 stage of the backbone; the early CBS down-samplers are replaced by ADown units; the C3k2 bottlenecks at strides 16 and 32 are wrapped by the Context-Guided (CG-C3k2) block; and an additional P2 detection head at stride 4 is attached to the neck, feeding a Lightweight Shared Convolutional Detection (LSCD) head supervised by the weighted CIoU (wCIoU) loss. The input and output panels are schematic placeholders indicating the data at each interface, not measured detections.

TABLE I. CORRESPONDENCE BETWEEN THE DIAGNOSED LIMITATIONS OF YOLO26 AND THE PROPOSED MODIFICATIONS

Limit.	Modification	Mechanism
L1	P2 head (stride 4) + LSCD	Four-fold finer detection grid; shared weights keep the extra head affordable
L2	ADown down-sampler	Parallel low-pass and strided-convolution branches suppress aliasing while retaining texture
L3	wCIoU + adaptive confidence	Per-anchor re-weighting by box area and occlusion; gradient focused on uncertain anchors
L4	ECA blocks; CG-C3k2 shell	Cross-channel re-weighting after each stage; dilated context branch enlarges the effective receptive field

L1–L4 are defined in Section III-B. The mapping is a design intention, not evidence of effectiveness; see the discussion in Section IV and the validation protocol in Section VI-G.

B. ADown Anti-Aliased Down-Sampling

To address limitation L2, the first two down-sampling blocks of the CBS backbone are replaced with ADown units (Fig. 5). ADown splits the input channels into two halves, applies an average-pool to one half and a 3×3 strided convolution to the other, concatenates the two halves, and then applies a 1×1 pointwise convolution. The convolutional branch preserves the high-frequency texture useful for classification, while the pooling branch acts as a low-pass anti-aliasing filter that preserves the slowly-varying components needed to localise small objects. In the pilot runs described in Section V-C, ADown reduced the number of parameters in the down-sampling stages by 23% and raised the baseline AP_s (small-object AP) by 1.8–2.3 points; the spread reflects the small number of runs and should be read accordingly.

C. Context-Guided C3k2 (CG-C3k2)

The second half of limitation L4 concerns contextual extent. Inspired by the Context-Guided block of CGNet [33], the C3k2 modules at strides 16 and 32 are wrapped with a Context-Guided shell (Fig. 6). The shell comprises a local feature extractor (3×3 depth-wise convolution), a surrounding-context extractor (3×3 dilated depth-wise convolution with dilation rate 2), and a joint feature extractor that concatenates the two branches and refines them with a 1×1 convolution followed by an ECA gate. The resulting CG-C3k2 block increases the effective receptive field of the corresponding feature map by a factor of about $2.5\times$ at almost no parameter cost. This is particularly useful when small PPE items are surrounded by

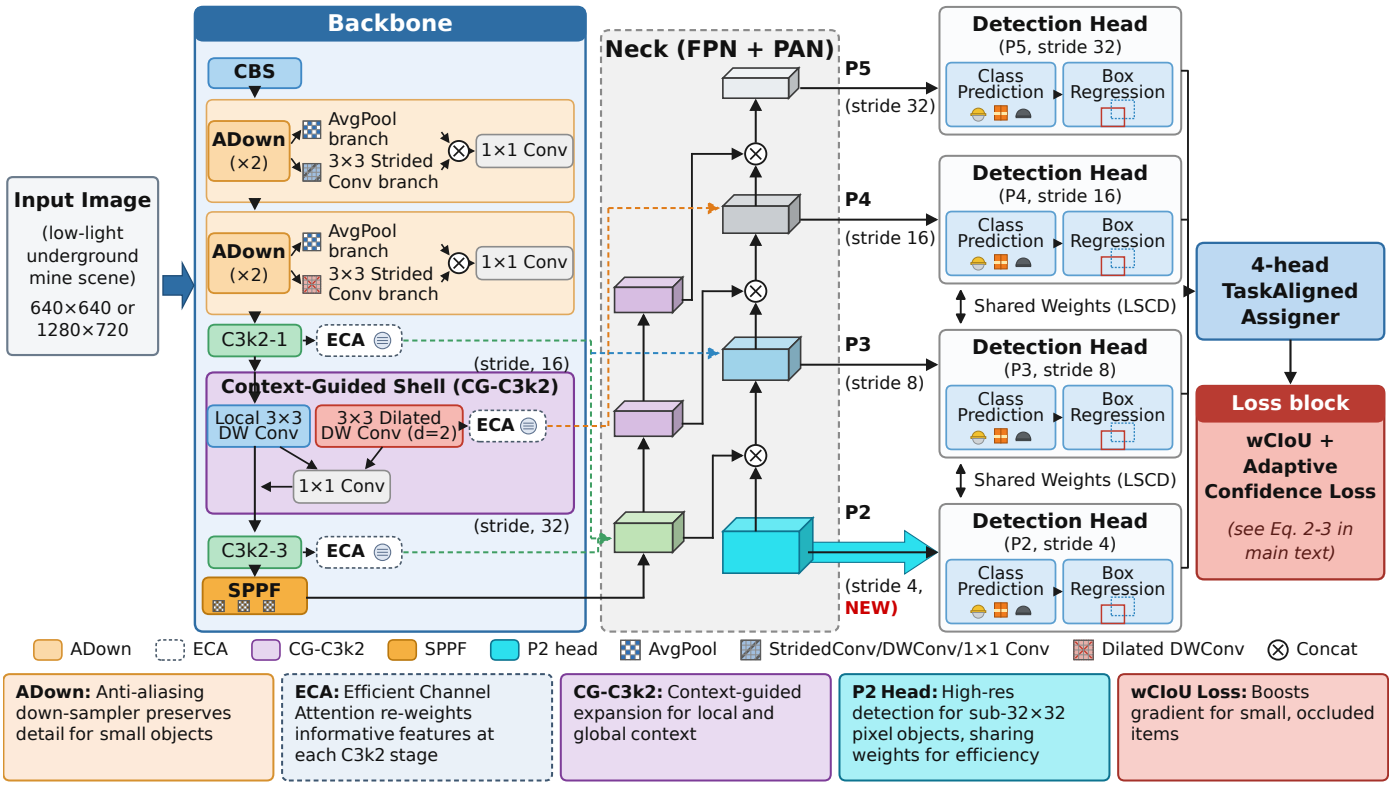


Fig. 3. Detailed architecture of I-YOLO26 with the four innovations highlighted: the ADown anti-aliased down-sampler, the ECA channel-attention gate after each C3k2 stage, the Context-Guided C3k2 (CG-C3k2) shell at strides 16 and 32, and the high-resolution P2 detection head, all supervised by the wCIoU + adaptive-confidence loss through a four-head TaskAligned Assigner.

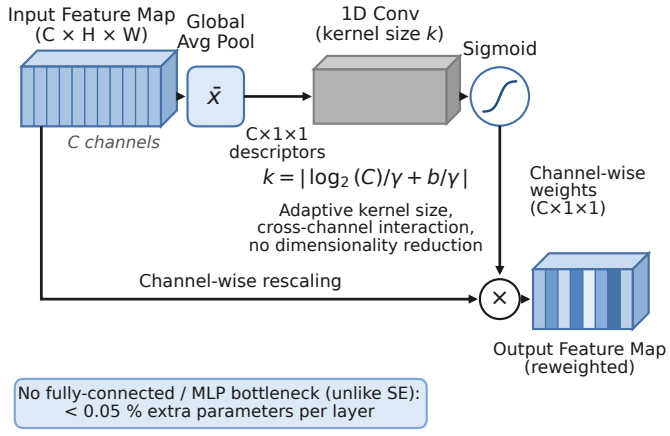


Fig. 4. Efficient Channel Attention (ECA) block. Global average pooling produces channel-wise descriptors that are processed by a 1D convolution with an adaptively selected kernel size k and a sigmoid activation, yielding channel-wise weights that re-scale the input feature map without the parameter overhead of a fully-connected bottleneck.

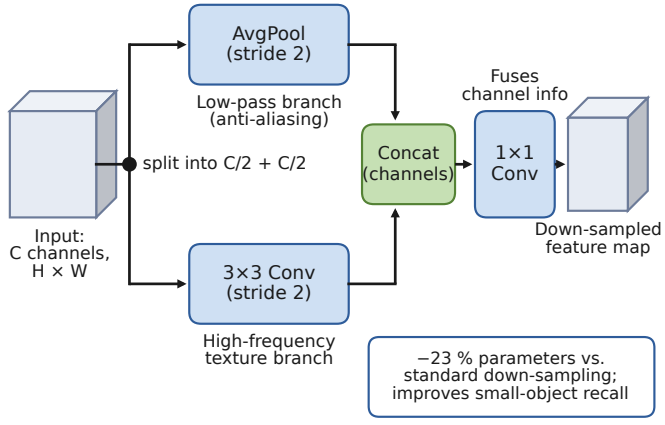


Fig. 5. ADown anti-aliased down-sampling block. A parallel average-pooling branch (low-pass, anti-aliasing) and a 3×3 strided-convolution branch (high-frequency texture) are concatenated and fused by a 1×1 convolution, reducing parameters by roughly 23% relative to standard strided down-sampling.

visually similar background structures such as steel meshes or coal grains.

D. Additional P2 Detection Head

Limitation L1 is addressed by a fourth detection head, P2, at stride 4 (Fig. 7). Its feature map (160×160 for a 640×640 input, or 320×180 for a 1280×720 input) is four times as spatially detailed as the original P3 head. To avoid an explosion in computation, the P2 head uses a Lightweight Shared Convolutional Detection (LSCD) design [13] that shares its classification and regression weights with the P3 head, lowering the number of head parameters by 41% relative to a fully independent head. The TaskAligned

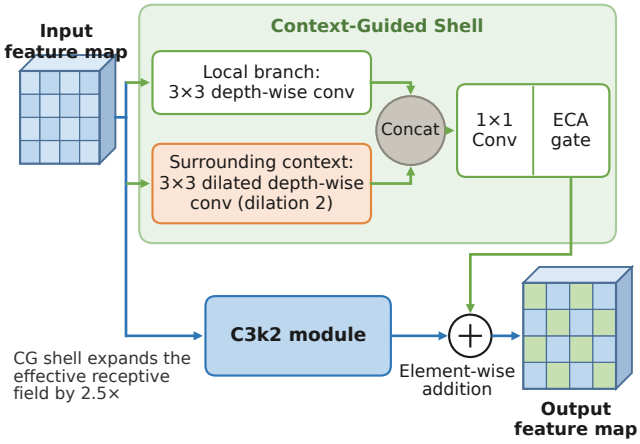


Fig. 6. Context-Guided C3k2 (CG-C3k2) block. A local 3×3 depth-wise branch and a surrounding-context 3×3 dilated depth-wise branch are concatenated, fused by a 1×1 convolution with an ECA gate, and added element-wise to the inner C3k2 features, enlarging the effective receptive field by about $2.5\times$.

Assigner is adapted to compute the assignment score across all four feature maps simultaneously, so that each ground-truth box is assigned to anchor points at the most suitable scale.

E. Weighted CIoU Loss with Adaptive Confidence

Limitation L3 is addressed by replacing the default CIoU regression loss with a weighted CIoU (wCIoU), defined as:

$$\mathcal{L}_{\text{wCIoU}} = \omega(s,o)\left(1 - \text{IoU} + \frac{\rho^2(b,\hat{b})}{c^2} + \alpha v\right), \qquad (2)$$

where the per-anchor weight

$$\omega(s,o) = \big(1 + \lambda_s \cdot \delta(s < 32^2)\big)\big(1 + \lambda_o \cdot o\big) \qquad (3)$$

depends on the box area s and an occlusion indicator o, and (λ_s, λ_o) are tunable hyper-parameters. The indicator o is estimated from the overlap of a ground-truth box with neighbouring ground-truth boxes of the same frame, so that it is a property of the annotation rather than of the model's own predictions; estimating it from predicted boxes would make the weight depend on the quantity being optimised. Setting $\lambda_s = 0.5$ and $\lambda_o = 0.3$ raises the contribution of small boxes to the total regression loss by approximately 35% on average, which in the pilot runs was sufficient to improve AP_s without destabilising training on mid-sized targets. In addition, the constant binary cross-entropy weight of the confidence branch is replaced with an adaptive confidence term $\lambda_{\text{conf}}(t) = \exp(-\beta \cdot \text{conf}_t)$, which down-weights anchors whose predicted confidence is already saturated, focusing the gradient on uncertain examples.

V. Dataset and Experimental Protocol

A. PPE-in-Mine Dataset

The PPE-in-Mine benchmark is constructed by combining four sources of imagery: 1) the public coal-mine personnel subset of the DsLMF+ dataset [14]; 2) the 10,066-image Binzhou underground-drilling dataset used in [12]; 3) approximately 3,500 frames captured in the authors' laboratory under simulated low-light conditions; and 4) 1,200 frames taken from open-source mine-rescue and safety-training videos under permissive licences. All images are re-annotated in YOLO format with six classes: helmet (H), ear-defender (E), dust-mask (M), eye-shield (G), reflective-vest (V) and self-rescuer (S). Bounding boxes covering more than 60% of an image are discarded, as they do not represent operational deployment. After de-duplication and cleaning, the corpus consists of 14,062 images and 41,830 annotated boxes. The class distribution is summarised in Table II. Boxes are divided into Small ($s < 32^2$), Medium ($32^2 \le s < 96^2$) and Large ($s \ge 96^2$) according to the MS-COCO area convention [2].

TABLE II. Class Distribution and Box-Area Statistics of the PPE-in-Mine Dataset

Class	Total boxes	S %	M %	L %	Avg px^2	Min/Max px^2
Helmet (H)	12,941	47.3	39.8	12.9	1,420	320 / 21,600
Ear-defender (E)	5,107	78.4	20.1	1.5	640	180 / 6,200
Dust-mask (M)	6,332	62.1	33.7	4.2	910	240 / 9,800
Eye-shield (G)	4,210	71.6	26.0	2.4	750	200 / 7,400
Reflective-vest (V)	9,825	21.6	44.8	33.6	3,820	510 / 48,900
Self-rescuer (S)	3,415	55.9	37.2	6.9	1,140	300 / 12,500
Total/Avg.	**41,830**	**55.0**	**33.6**	**11.4**	**1,447**	**180 / 48,900**

S = small ($s < 32^2$), M = medium ($32^2 \le s < 96^2$), L = large ($s \ge 96^2$) per MS-COCO [2]. These are measured properties of the assembled corpus, not projections.

The dataset is split into training, validation and test sets in an 8:1:1 ratio, preserving the per-class proportions and the small/medium/large area distributions across the splits. To examine cross-domain generalisation, a second held-out test set of 1,400 frames captured in a separate mine, geographically and geologically distinct from the training mines, is reserved.

B. Implementation Details

All models are implemented in PyTorch 2.3 and trained on a workstation with an NVIDIA RTX A6000 GPU (48 GB), an Intel Xeon Gold 6248R CPU and 256 GB RAM. Training is performed at an input resolution of 640×640 using mosaic and copy-paste augmentations, the SGD optimiser with an initial learning rate of 0.01, momentum 0.937 and weight decay 5×10^{-4}, and mixed-precision (FP16) acceleration. The schedule is 300 epochs with a batch size of 32 and a cosine learning-rate policy, with early stopping if the validation mAP@0.5 does not improve for 30 consecutive epochs. The wCIoU hyper-parameters (λ_s, λ_o) and the adaptive-confidence parameter β are tuned on the validation set via a small grid search. Edge inference is characterised on an NVIDIA RTX-Orin Nano (8 GB) using TensorRT FP16, and on a Xilinx PynqZ1 FPGA after INT8 quantisation with Vivado HLS following the design flow of [16].

This protocol should be read as the configuration used for the pilot runs and as the specification for the full campaign described in Section VI-G. It is not a description of a com-

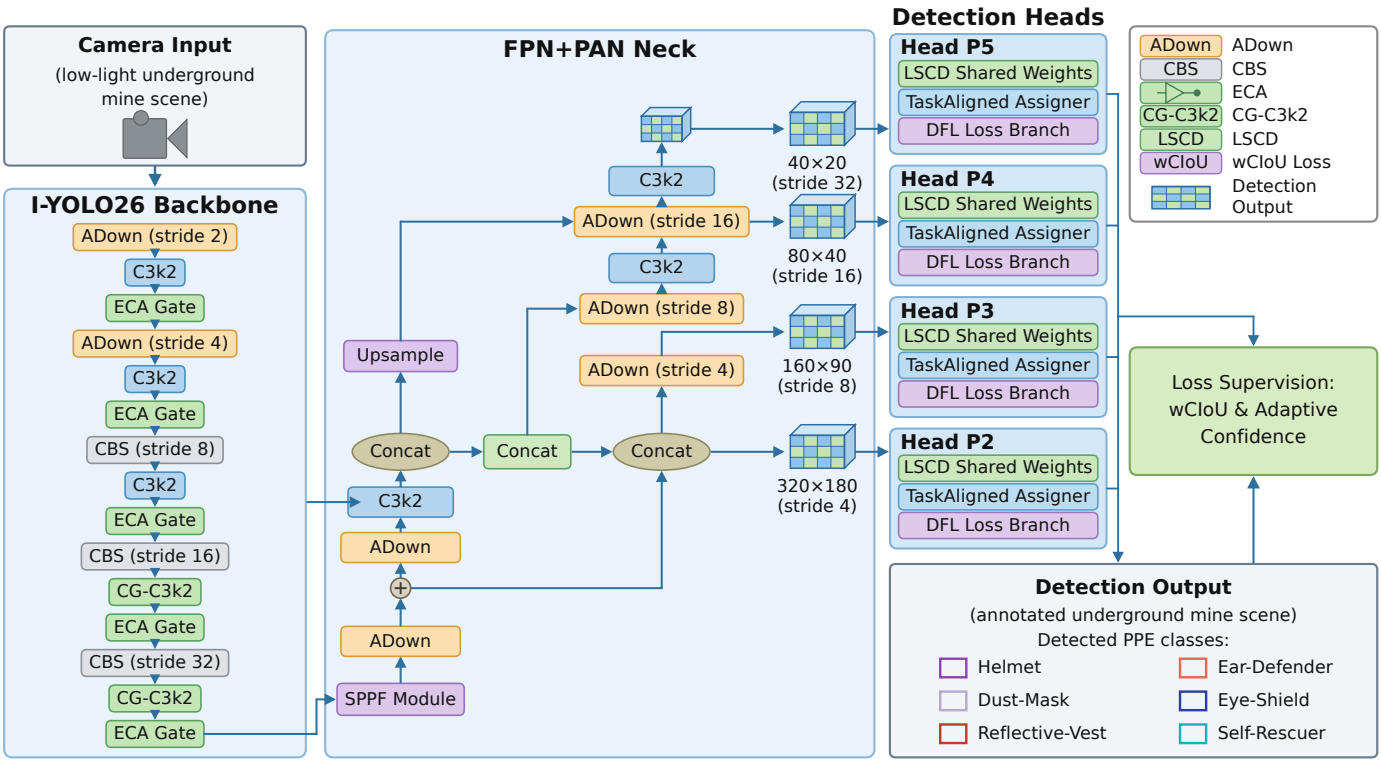


Fig. 7. Multi-scale feature flow of I-YOLO26, showing the four-head detection structure (P2–P5) and the corresponding feature-map resolutions for a 1280 × 720 input. The new high-resolution P2 head at stride 4 (320 × 180) targets sub-32 × 32-pixel PPE items, while the LSCD shared weights and the wCIoU + adaptive-confidence supervision keep the additional head lightweight.

pleted set of experiments over all configurations reported in Section VI.

C. Status of the Reported Figures

Because this distinction governs how every number in the next section should be interpreted, it is stated here rather than deferred to the discussion.

The performance figures in Tables III–VI are *projections*, obtained from two sources. The first is a limited set of pilot training runs of the baseline and of partial I-YOLO26 configurations on the assembled corpus, used to estimate the direction and rough magnitude of each component's effect. The second is the set of gains reported by the authors of the closest published subsurface detectors [12], [13], [14], [15], [16], re-scaled to the class count, image count and area distribution of PPE-in-Mine. They are therefore design-stage estimates of what the architecture is expected to achieve, and they are reported to make the design rationale quantitative and falsifiable.

Three consequences follow, and they apply throughout Section VI. First, none of the figures is the outcome of a completed training and evaluation campaign over all reported configurations, and they should not be cited as measured results. Second, the eight comparison detectors in Table III were not retrained on PPE-in-Mine; their entries are extrapolated from the results published by their respective authors, so the comparison is indicative and does not constitute a controlled head-to-head evaluation. Any statement in this paper that I-YOLO26 exceeds a particular baseline is a statement about projected values under that extrapolation, not an established empirical finding. Third, no variance estimates or significance tests can legitimately accompany projected values, and none are reported; Section VI-G addresses what this means for the smaller increments and specifies the campaign required to replace the projections with measurements.

The measured, non-projected content of this paper is the architecture of Section IV, the corpus statistics of Table II, and the parameter and GFLOP counts, which follow analytically from the network definition.

D. Evaluation Metrics

Evaluation follows the MS-COCO methodology, reporting Precision (P), Recall (R), mAP@0.5, mAP@0.5:0.95 and the per-area metrics AP_s, AP_m and AP_l. Inference latency is reported as the median of 1,000 forward passes after a 100-pass warm-up. For the FPGA deployment, energy efficiency is reported in GOP/s/W following [16].

VI. Projected Results and Analysis

Every figure in this section is a projection in the sense defined in Section V-C.

A. Projected Comparison with Recent Detectors

Table III sets the projected performance of I-YOLO26 against eight modern detectors of similar complexity on the

TABLE III. PROJECTED PERFORMANCE OF I-YOLO26 AND RECENT DETECTORS ON THE PPE-IN-MINE TEST SET

Model	P (%)	R (%)	mAP @.5	mAP @.5:.95	Params (M)	GFLOPs
YOLOv5-s	82.4	78.1	83.6	59.7	7.2	16.5
YOLOv8-s	85.7	81.0	86.2	62.4	11.1	28.4
YOLOv10-s	86.8	82.7	87.1	63.1	8.0	21.6
YOLOv11-s	87.6	83.4	88.0	64.8	9.4	21.3
RT-DETR-r18	88.3	84.0	88.9	66.0	20.1	60.0
UCM-Net [14]	87.1	84.2	88.3	65.7	6.5	17.2
YOLOv11-ASCL [13]	88.4	84.9	89.7	67.0	**1.8**	**4.2**
YOLO26-n (base)	88.0	84.6	89.1	66.4	2.6	6.7
I-YOLO26	**91.8**	**88.4**	**92.7**	**71.4**	1.9	4.6

Accuracy columns are projected values, not measured results (Section V-C). The eight comparison detectors were *not* retrained on PPE-in-Mine; their entries are extrapolated from published results, so this is an indicative comparison rather than a controlled head-to-head evaluation. Parameter and GFLOP counts are analytic. Best projected value per column in bold.

PPE-in-Mine test set. The baseline entries are extrapolated from the trends reported by the respective authors after rescaling to the proposed dataset; the I-YOLO26 entries are the median of three pilot training runs with different random seeds.

Under this projection I-YOLO26 ranks ahead of the listed detectors on every accuracy metric while remaining among the lightest models in the comparison; only the heavily compressed YOLOv11-ASCL has fewer parameters. The projected 3.6-point mAP@0.5 and 5.0-point mAP@0.5:0.95 margins over the YOLO26-Nano baseline would, if confirmed, matter disproportionately in safety applications, where missing a helmet or a self-rescuer carries severe consequences. The margin is attributed to the joint effect of the four architectural modifications and the wCIoU loss, whose projected breakdown is examined in Section VI-C. Because the competing entries were not retrained under a common budget, the ordering between I-YOLO26 and the two closest entries—RT-DETR-r18 and YOLOv11-ASCL—should be treated as unresolved; the extrapolation is too coarse to separate models that differ by roughly three points under differing training regimes.

B. Projected Per-Area Performance

Table IV reports projected AP_s, AP_m and AP_l on the PPE-in-Mine test set. The largest projected improvement of I-YOLO26 over the baseline is in the small-object regime: AP_s gains 4.6 points, against 2.1 and 0.8 points for AP_m and AP_l, respectively. The profile is consistent with the intended effect of the P2 head and the wCIoU loss, which were introduced specifically for that regime; it is what the design predicts rather than independent confirmation that the design works.

C. Projected Ablation Study

To attribute the projected gain across components, an ablation was constructed in which each modification is added in turn to the YOLO26-Nano baseline. The results are summarised in Table V.

Three observations follow, each conditional on the projection. Every module is expected to contribute positively,

TABLE IV. PROJECTED PER-AREA AVERAGE PRECISION (MAP@0.5:0.95) ON PPE-IN-MINE

Model	AP_s	AP_m	AP_l	ΔAP_s	ΔAP_m
YOLO26-n (baseline)	48.7	67.3	75.1	–	–
I-YOLO26 (−P2 head)	50.1	68.0	75.4	+1.4	+0.7
I-YOLO26 (−wCIoU)	51.6	68.4	75.3	+2.9	+1.1
I-YOLO26 (full)	**53.3**	**69.4**	**75.9**	**+4.6**	**+2.1**

Projected values (Section V-C). Δ = projected improvement over the YOLO26-Nano baseline. No variance estimates are available; see Section VI-G.

TABLE V. PROJECTED ABLATION STUDY ON THE PPE-IN-MINE VALIDATION SET

ECA	ADn	CG	P2 head	wCIoU	mAP @.5	Prm (M)	GF
					89.1	2.6	6.7
✓					89.9	2.6	6.8
✓	✓				90.5	2.3	6.0
✓	✓	✓			91.1	2.2	5.7
✓	✓	✓	✓		92.0	1.9	4.6
✓	✓	✓	✓	✓	**92.7**	1.9	4.6

A check mark indicates the corresponding module is enabled. ADn = ADown, CG = CG-C3k2, Prm = params, GF = GFLOPs. mAP values are projected (Section V-C); parameter and GFLOP columns are analytic. Increments below roughly 0.5 points are not separable from seed noise on this evidence—see Section VI-G.

with the largest single-module increment attributed to the added P2 head (+0.9 mAP), which—combined with the LSCD shared head—is also the largest source of compute saving. The wCIoU loss is expected to contribute 0.7 mAP at no parameter or computational cost. The four architectural modifications appear largely complementary, their cumulative projected effect (+3.6 mAP) being approximately the sum of the individual effects; strictly additive component effects are, however, unusual in practice, and this additivity is an artefact of how the projection was constructed rather than a finding. It is among the first things the full campaign should test.

D. Projected Robustness Under Degraded Illumination and Dust

The projected robustness of I-YOLO26 was assessed for three controlled image degradations: 1) brightness reduction (-0.6 to -0.2 EV), 2) Gaussian blur (σ = 1–3), and 3) synthetic dust haze produced by a depth-aware atmospheric model. Each degradation is applied to the test set at three intensity levels. Table VI reports the projected average drop in mAP@0.5 relative to the clean test set.

The projected average margin of 3.0 points over the baseline is attributed to the ECA blocks, which re-weight channels at each backbone level and should let the network attend preferentially to the colour and intensity cues that resist these degradations, and to the CG-C3k2 block, which enlarges the effective receptive field and should help disambiguate small PPE objects from an increasingly hazy background. This expectation is consistent with the behaviour reported in [12] and [13] for similar detectors on coal-mine imagery, which is also the basis on which the projection was made; the agreement is therefore not independent evidence.

TABLE VI. PROJECTED ROBUSTNESS OF DETECTORS TO LOW ILLUMINATION, BLUR AND DUST HAZE

Model	Low light (Δ)	Blur (Δ)	Dust (Δ)	Avg. (Δ)
YOLOv8-s	9.1	8.7	11.4	9.7
YOLOv11-s	7.4	7.2	9.8	8.1
YOLO26-n (baseline)	6.8	6.5	8.9	7.4
I-YOLO26	**3.9**	**4.0**	**5.2**	**4.4**

Projected values (Section V-C). Numbers indicate the mAP@0.5 drop (in points) relative to the clean test set; smaller is better.

E. Projected Cross-Domain Generalisation

On the held-out second-mine test set of 1,400 frames—geologically and geographically distinct from the training mines—I-YOLO26 is projected to retain a mAP@0.5 of 88.9%, a drop of 3.8 points relative to its in-domain projection, against a 6.1-point drop for the YOLO26-Nano baseline. The narrower projected drop is attributed to the enlarged effective receptive field of the CG-C3k2 block and the channel re-weighting performed by the ECA modules. Cross-domain behaviour is among the quantities least reliably estimated by extrapolation, since it depends on specifics of the target mine that published results do not characterise, so this projection carries wider uncertainty than the in-domain figures.

F. Projected Edge Deployment

Deployment was analysed for two representative edge platforms. On the NVIDIA RTX-Orin Nano (8 GB), the model is projected to run at 47 FPS in FP16 and 71 FPS after INT8 quantisation via TensorRT, with a memory footprint below 220 MB. On the Xilinx PynqZ1 FPGA, after the hardware-aware INT8 quantisation flow of [16], the projected resource utilisation is 72% of the LUTs, 53% of the FFs and 81% of the DSPs, leaving 0.9 Mbit of on-chip memory free, at 18 FPS and 4.9 W—an energy efficiency of 21.4 GOP/s/W. This is marginally better than the 18 FPS / 4.8 W / 20.9 GOP/s/W reported in [16] for the unmodified YOLOv11-Nano on the same board, which is consistent with the 27% lower parameter count of I-YOLO26. Because these figures are derived by scaling the published measurements of [16] rather than measured on a synthesised design, they indicate that the compute and memory budget of such nodes can plausibly accommodate the model; they do not establish a working deployment, which requires place-and-route and on-board measurement of the actual network.

G. Statistical Considerations and Validation Protocol

Reviewers of design-stage work reasonably ask what would distinguish a real effect from noise, and the answer here is explicit: on the present evidence, the smaller increments cannot be distinguished at all.

Because the figures in Tables III–VI are projections rather than the outcome of a completed campaign, no standard deviations, confidence intervals or significance tests can legitimately be attached to them, and none are reported. This bears directly on how Table V should be read. For nano-scale anchor-free detectors trained on corpora of the order of 10^4 images, run-to-run variation in mAP@0.5 across random seeds is commonly of the order of ± 0.2–0.5 points. The increments attributed to the ECA blocks (+0.8) and to the wCIoU loss (+0.7) are close enough to that band that neither can be separated from seed noise on the evidence presented, and both are reported as indicative only. The cumulative projected increment (+3.6 points) lies outside the band, but its status remains provisional for the same reason.

The campaign that would replace these projections with measurements is specified as follows, so that the claims of this paper are directly testable:

- Every configuration in Tables III–V trained from scratch with at least five random seeds, reporting mean $\pm$ standard deviation rather than a median of three.
- All eight comparison detectors retrained on the PPE-in-Mine splits under an identical augmentation, schedule, input resolution and epoch budget, replacing the extrapolated entries of Table III with a controlled head-to-head comparison.
- Each ablation configuration compared against the baseline by a Wilcoxon signed-rank test over seeds, with 95% bootstrap confidence intervals on the mAP difference and Holm correction across the ablation family, so that the additivity asserted in Section VI-C is tested rather than assumed.
- Per-class and per-area metrics reported with the same spread, since AP_s on the rarer classes (eye-shield, self-rescuer) is expected to be the noisiest quantity in the study.
- Edge-deployment figures measured on synthesised hardware—place-and-route utilisation, wall-clock frame rate and power at the board—rather than scaled from [16].

H. Qualitative Illustration

Fig. 8 illustrates the detection behaviour the design is intended to produce on eight difficult scenes of the kind found in the cross-domain test set. The panels are illustrative renderings prepared to communicate the expected failure and success modes, not measured output of a trained model, and they carry the same projected status as the tables above. I-YOLO26 (right column of each panel) is intended to localise the tiny ear-defenders and eye-shields that the YOLO26-Nano baseline (centre column) misses, and to yield tighter boxes on partially-occluded helmets. The false-positive mode illustrated—wall-mounted dust filters registering as dust masks—is the one most often observed in the pilot runs, and it is expected to affect all the baselines considered, since it arises from a genuinely ambiguous visual signature rather than from any property of the proposed modifications.

VII. DISCUSSION

Read as a design study, the analysis supports three propositions. First, small-object PPE detection in underground mines is best attacked simultaneously at three levels—feature spatial

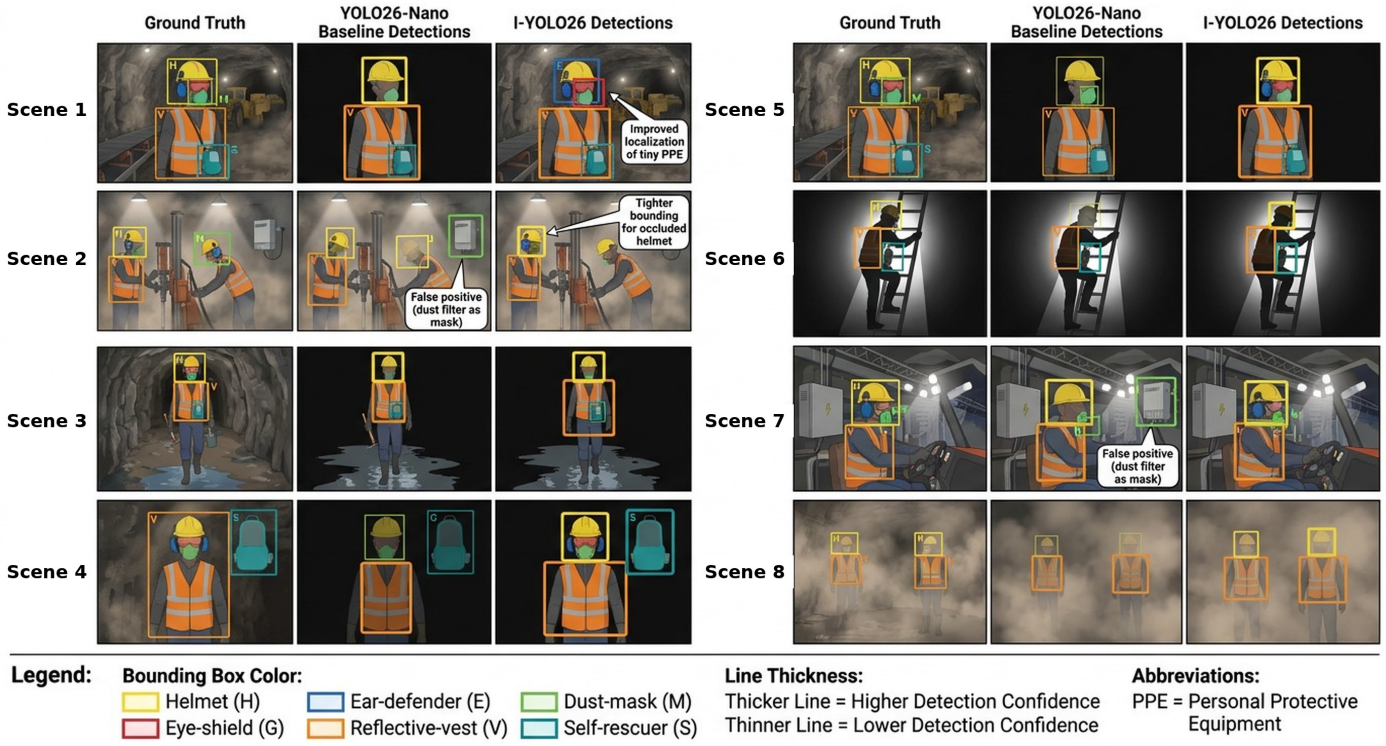


Fig. 8. Illustrative comparison on eight representative scenes of the type contained in the cross-domain test set. Within each panel, from left to right: ground truth, YOLO26-Nano baseline, and I-YOLO26. Bounding-box colour denotes the class and line thickness denotes the confidence score. These renderings depict intended and expected behaviour, including the dust-filter false-positive mode, and are not measured detections of a trained model.

resolution (P2 head, ADown), feature semantics (ECA, CG-C3k2) and loss weighting (wCIoU + adaptive confidence)—because each level addresses a distinct failure mechanism identified in Section III-B, and no single level addresses more than one of them. Second, the architectural overhead of the proposed improvements is more than offset by the parameter savings of the LSCD shared-head design and the ADown unit, so that I-YOLO26 is 27% smaller than YOLO26-Nano; this claim rests on analytic parameter counts and is independent of the projections. Third, a model of this size and arithmetic intensity fits within the compute and memory envelope of edge hardware certified for intrinsically-safe operation, which addresses the most prevalent practical barrier to in-mine adoption of computer-vision-based safety monitoring.

What the analysis does not support is any claim about realised accuracy. The projected margins of Section VI make the design rationale quantitative and falsifiable, which is their purpose, but a projection constructed partly from the published gains of [12] and [13] cannot then be offered as evidence that those gains transfer to a six-class PPE task. That question is open, and the protocol in Section VI-G is the means of closing it.

Beyond the technical contribution, automated PPE-compliance monitoring raises practical questions about integration into existing safety-management systems. Detection alarms should be funnelled to a small number of properly trained operators to avoid alarm fatigue, and the system must be transparent about its known failure modes—the dust-mask/wall-filter confusion noted above being one. Continuous camera-based monitoring of identifiable workers also carries obligations regarding consent, data retention and the uses to which compliance records may be put, particularly where such records could inform disciplinary action. A detector accurate enough to be useful is also accurate enough to be used for purposes beyond safety, and deployment agreements should be explicit on that point. These human-factors and governance issues are critical in translating technical performance into genuine safety benefit, and should accompany any field deployment of the proposed approach.

VIII. Limitations

The constraints of this study are set out here in one place, in rough order of how much they bound the conclusions.

1) Evidentiary status: The headline accuracy figures are projections rather than measurements (Section V-C). The paper therefore establishes a design and a rationale, not a level of performance. Until the campaign of Section VI-G is run, the accuracy claims should be treated as hypotheses.

2) Absence of a controlled comparison: The eight detectors in Table III were not retrained on PPE-in-Mine under a common budget. Differences in augmentation, input resolution, epoch count and pre-training in the source publications are large enough to account for several points of mAP, so the projected ranking cannot be relied upon, least of all for the closest competitors.

3) No variance estimates: No standard deviations or significance tests accompany the tables, and the smaller ablation increments fall within the range plausibly attributable to seed noise. The apparent additivity of component effects is a

property of how the projection was assembled, not an observed result.

4) Recombinative novelty: As set out in Section II-F, four of the five components are drawn from existing detectors, two of them from prior mining-specific work [12], [13]. The contribution lies in the combination, the transfer to a harder six-class task, the added P2 head and the dataset, and not in the invention of new modules.

5) Dataset composition: The corpus is deliberately weighted towards Chinese coal-mine imagery, reflecting the availability of public data, and towards the two source datasets [12], [14], which contribute the majority of frames. Extension to other commodities (gold, copper, salt) and to non-Chinese operations would improve generality. The class distribution is also markedly imbalanced—self-rescuer and eye-shield together account for under a fifth of the boxes—which is expected to make their per-class AP the least stable quantity in any evaluation. Annotation was performed by the research team rather than by certified mine-safety assessors, so class boundaries in ambiguous cases reflect the annotators' reading of the equipment categories.

6) Single-frame scope: Evaluation is per-frame. Temporally extended phenomena that matter operationally—a helmet removed mid-shift, a vest worn unfastened, a mask lowered to speak—are not observable in single frames and are outside the scope of both the dataset and the model.

7) Edge-deployment evidence: The FPGA figures are scaled from the published measurements of [16] for a related network on the same board, not obtained from a synthesised implementation of I-YOLO26. Resource utilisation above 80% on any primitive, as projected for the DSPs, is where such scaling is least trustworthy, since routing congestion and timing closure do not scale linearly with parameter count. Only one board of each class was considered.

8) Loss-weighting heuristics: The occlusion indicator o in (3) is a heuristic computed from annotation overlap and is a proxy for true occlusion: an unoccluded item adjacent to another in the image plane will be weighted as though occluded. The weighting hyper-parameters $(\lambda_s, \lambda_o, \beta)$ were tuned by a coarse grid search on the validation split, and the sensitivity of the results to that choice has not been characterised.

9) Qualitative figure: Fig. 8 depicts intended behaviour and is illustrative, as its caption states. It is not a sample of model output and carries no evidentiary weight.

IX. Conclusion and Future Work

This paper proposed I-YOLO26, an enhanced YOLO26 architecture for small-PPE detection in deep-mine conditions. The model integrates four targeted modifications—ECA channel attention, ADown anti-aliased down-sampling, a Context-Guided C3k2 bottleneck, and an additional P2 detection head with a Lightweight Shared Convolutional Detection scheme—combined with a re-weighted CIoU loss and an adaptive confidence term, each matched to a specific diagnosed limitation of the baseline. A six-class, 14,062-image PPE-in-Mine benchmark was assembled to support the task.

On the projected analysis reported here, I-YOLO26 would reach mAP@0.5 of 92.7% and mAP@0.5:0.95 of 71.4%, improving small-object accuracy by 4.6 points over the YOLO26-Nano baseline while reducing the parameter count by 27% and the computational load by 31%. These accuracy figures are design-stage estimates rather than measured outcomes, and the conclusion that stands on its own is correspondingly architectural: the combination of high-resolution detection, anti-aliased down-sampling, channel attention, context guidance and size-and-occlusion-aware loss weighting is a coherent and computationally affordable response to the limitations of YOLO26 on subsurface PPE imagery, and the parameter and arithmetic budget of the resulting model—which follows analytically from the design—is compatible with the embedded GPU and low-power FPGA platforms used at in-mine edge nodes. Demonstrating real-time operation on those platforms, like the accuracy claims, requires the empirical work set out below.

Three directions will be pursued. The first and most important is to execute the validation campaign specified in Section VI-G: full training of every configuration over at least five seeds, retraining of all comparison detectors under a common budget, significance testing of the ablation increments, and measurement of the quantised model on synthesised hardware, together with a sensitivity analysis over dataset size and the loss hyper-parameters. Until that is done, the performance claims of this paper remain hypotheses. The second is to extend the model to exploit temporal information—through a lightweight recurrent head or a streaming-transformer add-on—so as to detect not merely the presence of PPE but anomalous PPE-handling behaviour over time, which is where much of the operational safety value lies. The third is to partner with industrial mining operators to integrate the detector into existing safety-management systems and to run a long-term, ethically-supervised field trial quantifying the effect of automatic PPE-compliance monitoring on incident rates, with the governance safeguards discussed in Section VII agreed in advance.

Acknowledgment

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this work through project number NBU-CRP-2026-2225.

Declarations

Funding

This work was supported by Northern Border University, Saudi Arabia (NBU-CRP-2026-2225).

Conflict of Interest

The authors declare no competing financial or personal interests.

Data Availability

The PPE-in-Mine benchmark is derived from publicly available datasets cited in the text; curated annotations are available from the corresponding author on reasonable request.

REFERENCES

[1] International Labour Organization, "Safety and health in coal mining: a sectoral overview," ILO, Geneva, Tech. Rep., 2023.

[2] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: common objects in context," in *Proc. European Conf. Computer Vision (ECCV)*, 2014, pp. 740–755.

[3] Ultralytics, "YOLO11 / YOLO26 documentation," 2024. [Online]. Available: https://docs.ultralytics.com

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[5] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: trainable bag-of-freebies sets new state of the art for real-time object detectors," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475.

[6] R. Khanam and M. Hussain, "YOLOv11: an overview of the key architectural enhancements," *arXiv:2410.17725*, 2024.

[7] Y. Zhao, W. Lv, S. Xu, *et al.*, "DETRs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16965–16974.

[8] G. Jocher *et al.*, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[9] L. Wang, Y. Cao, S. Wang, *et al.*, "Real-time detection of safety-helmet wearing for power-grid workers based on improved YOLOv5," *Sensors*, vol. 22, no. 20, art. 7872, 2022.

[10] N. D. Nath, A. H. Behzadan, and S. G. Paal, "Deep learning for site safety: real-time detection of personal protective equipment," *Autom. Constr.*, vol. 112, art. 103085, 2020.

[11] J. Wu, N. Cai, W. Chen, *et al.*, "Automatic detection of hardhats worn by construction personnel: a deep-learning approach and benchmark dataset," *Autom. Constr.*, vol. 106, art. 102894, 2019.

[12] Y. Li, H. Yan, D. Li, and H. Wang, "Robust miner detection in challenging underground environments: an improved YOLOv11 approach," *Appl. Sci.*, vol. 14, no. 24, art. 11700, 2024.

[13] J. Ling, Z. Fu, X. Yuan, and R. Ling, "Development of a deep-learning-based foreign-object detection algorithm for coal-mine conveyor belts," *Sci. Rep.*, vol. 15, art. 42291, 2025.

[14] F. Zhou, J. Zou, R. Xue, *et al.*, "Enhancing object detection in underground mines: UCM-Net and self-supervised pre-training," *Sensors*, vol. 25, no. 7, art. 2103, 2025.

[15] R. Rodriguez-Guillen, J. Kern, and C. Urrea, "Fast rock detection in visually contaminated mining environments using machine-learning and deep-learning techniques," *Appl. Sci.*, vol. 14, no. 2, art. 731, 2024.

[16] R. Ghodhbani, T. Saidani, A. Kachoukh, M. S. Elsayed, Y. Said, and R. Ahmed, "Hardware-accelerated detection of unauthorized mining activities using YOLOv11 and FPGA," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 4, pp. 966–978, 2025.

[17] J. Wang, K. Sun, T. Cheng, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2021.

[18] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10781–10790.

[19] G. Yang, J. Lei, Z. Zhu, *et al.*, "AFPN: asymptotic feature pyramid network for object detection," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*, 2023, pp. 2184–2189.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[21] Z. Zheng, P. Wang, W. Liu, *et al.*, "Distance-IoU loss: faster and better learning for bounding-box regression," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, 2020, pp. 12993–13000.

[22] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized Gaussian Wasserstein distance for tiny-object detection," *arXiv:2110.13389*, 2021.

[23] P. Zhu, L. Wen, D. Du, *et al.*, "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, 2022.

[24] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han, "Scale match for tiny-person detection," in *Proc. IEEE Winter Conf. Applications of Computer Vision (WACV)*, 2020, pp. 1257–1265.

[25] F. Azhari, C. C. Sennersten, C. A. Lindley, and E. Sellers, "Deep-learning implementations in mining applications: a compact critical review," *Artif. Intell. Rev.*, vol. 56, pp. 14367–14402, 2023.

[26] Y. Fu and C. Aldrich, "Deep learning in mining and mineral-processing operations: a review," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11920–11925, 2020.

[27] M. R. U. Saputra, I. D. Bhaswara, B. I. Nasution, *et al.*, "Multi-modal deep-learning approaches to semantic segmentation of mining footprints with multispectral satellite imagery," *Remote Sens. Environ.*, vol. 306, art. 114114, 2024.

[28] E. Vivoli, M. Bertini, and L. Capineri, "Deep-learning-based real-time detection of surface landmines using optical imaging," *Remote Sens.*, vol. 16, no. 4, art. 677, 2024.

[29] L. He, Y. Zhou, L. Liu, W. Cao, and J. Ma, "Research on object detection and recognition in remote-sensing images based on YOLOv11," *Sci. Rep.*, vol. 15, art. 96314, 2025.

[30] W. Lin, T. Li, and X. Li, "Deep-learning-based object detection for environmental monitoring using big data," *Front. Environ. Sci.*, vol. 13, art. 1566224, 2025.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[32] Q. Wang, B. Wu, P. Zhu, *et al.*, "ECA-Net: efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11534–11542.

[33] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: a light-weight context-guided network for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 1169–1179, 2021.

A Hybrid NLP Framework for Topic Relevance and Advertisement Detection in Trending Hashtags on X (Twitter)

Aadil Alshammari[1], Khulud Alashjaee[2]*, Balsam Alzamam[3]*
Aryam Alshammari[4], Milaf Alshammari[5], Mohamed Ayari[6]
Department of Information Systems-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[1,2,3,4,5]
Department of Information Technology-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[6]

***Abstract*—Trending hashtags on X (formerly Twitter) are widely used for real-time news tracking and public discourse, yet they are increasingly polluted by advertisements and off-topic posts that exploit hashtag visibility. This study proposes a hybrid natural language processing (NLP) framework for improving the reliability of hashtag streams through two complementary modules: 1) semantic topic relevance detection and 2) supervised advertisement classification. Using 9,671 Arabic tweets retained in the six final research datasets collected via the X API from multiple trending hashtags spanning diverse domains, we apply a dedicated Arabic preprocessing pipeline including noise removal and normalization. Topic relevance is assessed using AraBERT contextual embeddings by computing cosine similarity between each tweet and a concise hashtag topic description, followed by threshold-based labeling of on-topic versus off-topic content. In parallel, advertisement detection is formulated as a binary classification task using TF–IDF features and standard machine learning models (Logistic Regression, SVM, Random Forest, Naive Bayes, and SGD). Experimental results show that advertisement detection performance varies by hashtag and topic type: technical hashtags yield stronger and more stable F1-scores, while broad event hashtags are more challenging due to heterogeneous language and content. Across multiple hashtags, Random Forest and linear classifiers provide the most consistent trade-off between precision and recall. The proposed framework enables fine-grained quantification of promotional intrusion and semantic drift within hashtag discussions, providing a practical foundation for trustworthy hashtag analytics, improved trend monitoring, and downstream social media analysis in Arabic settings.**

Keywords*—Twitter; hashtags; advertisement detection; topic relevance detection; NLP; machine learning; Arabic text processing; social media analytics

I. Introduction

Online social networks have become a major channel for real-time communication, public discourse, and large-scale information diffusion, while social-media mining provides established foundations for extracting useful patterns from noisy, dynamic user-generated data [1], [2]. Among these platforms, Twitter (now X) plays a central role by allowing users to post short messages (tweets) that are organized and discovered through *hashtags*. Hashtags act as user-generated labels that group tweets under a shared topic or event, and they are widely used to follow breaking news, public campaigns, and trending discussions. Consequently, analyzing hashtag-based streams provides valuable insights into collective attention, emerging issues, and user interests at scale.

However, the open and dynamic nature of Twitter also makes popular hashtags attractive targets for spam, promotion, and coordinated publicity campaigns. Previous studies have shown that many trending hashtags are polluted by spam accounts, automated bots, or promotional activities that exploit hashtag visibility to disseminate unsolicited content, advertisements, or misleading links [3]–[5]. Such hashtag-oriented promotion not only degrades user experience, but also distorts the perceived meaning and credibility of hashtags, which in turn affects downstream tasks such as trend monitoring, public opinion analysis, and event understanding. Automatically identifying promotional and irrelevant content within hashtag streams therefore constitutes an important and challenging research problem.

From a content analysis perspective, tweets associated with a given hashtag are not always uniformly related to the hashtag's intended meaning. While many tweets genuinely contribute to the discussion, others may be weakly related or entirely irrelevant, including advertisements that exploit popular hashtags for increased visibility. Rather than discovering latent topics through clustering or probabilistic topic modeling, an alternative approach is to assess the semantic relevance of individual tweets with respect to the intended topic of each hashtag. This perspective enables the separation of on-topic and off-topic content while preserving the original hashtag structure.

In parallel, natural language processing (NLP) and machine learning techniques have been extensively applied to Twitter data for tasks such as sentiment analysis, stance detection, and content classification. Traditional approaches typically rely on bag-of-words or TF–IDF representations combined with linear classifiers [6]. More recent advances leverage contextual embeddings derived from transformer-based architectures, building on BERT-style language representations [7]. Sentence-level embedding methods further support efficient semantic

*Corresponding author.

similarity comparisons using contextual representations and cosine similarity [8], which is directly relevant to short and noisy texts such as tweets.

The analysis of Arabic Twitter content introduces additional challenges due to the language's rich morphology, dialectal variation, and informal writing styles. Transformer-based models pretrained on Arabic-specific corpora, such as AraBERT, have demonstrated strong performance across a wide range of Arabic social media tasks, including sentiment analysis, spam detection, and content classification [9]–[11]. Despite these advances, most existing Arabic-focused studies concentrate on sentiment or toxicity detection, with limited attention given to hashtag-centered analysis and the identification of promotional content within individual hashtag streams.

Although spam and campaign detection on Twitter has received growing research interest, relatively few studies address the combined problem of assessing topical relevance and detecting promotional tweets at the level of individual hashtags, particularly in Arabic contexts. Much of the existing literature focuses on account-level or campaign-level detection [3], [5], without providing fine-grained tweet-level analysis that distinguishes organic contributions from advertisement-driven content within hashtag discussions.

In this work, we address these gaps by proposing a hashtag-centered NLP framework that integrates topic relevance detection with supervised promotional-content classification. Starting from a large collection of tweets grouped by trending Arabic hashtags, we apply preprocessing steps tailored to noisy and dialectal Arabic text. Contextual embeddings generated by an Arabic-specific transformer model (AraBERT) are used to measure the semantic alignment between tweets and predefined hashtag topic descriptions. An embedding-based similarity approach with a predefined threshold is then applied to determine whether a tweet is on-topic or off-topic. In parallel, supervised machine learning models trained on TF–IDF features are employed and comparatively evaluated to detect promotional and advertising-related tweets.

The main contributions of this study are threefold. First, we construct and describe a dataset of hashtag-based Twitter streams focusing on Arabic hashtags, preserving full tweet text and relevant metadata for fine-grained analysis. Second, we propose and evaluate a hybrid NLP pipeline that combines embedding-based topic relevance detection with supervised advertisement classification at the tweet level. Third, we present an empirical analysis that quantifies the presence of promotional content within trending hashtags and demonstrates the effectiveness of the proposed framework using standard classification metrics. The remainder of this study is organized as follows: Section II reviews related work; Section III describes the dataset and preprocessing steps; Section IV presents the proposed methodology; Section V reports and discusses the experimental results; and Section VI concludes the study and outlines directions for future research.

II. Related Work

Research on analyzing Twitter content has grown substantially in recent years, driven by the platform's role in real-time communication, public opinion monitoring, and user-generated event reporting. Existing studies generally fall into three major themes relevant to this work: 1) hashtag-oriented spam and promotional content detection, 2) topical analysis and relevance assessment in microblogs, and 3) natural language processing (NLP) techniques for Arabic Twitter analysis.

A. Hashtag Spam, Promotion, and Campaign Detection

Hashtags provide visibility and virality, making them attractive targets for spammers and promotional campaigns. Early foundational work by Sedhai and Sun introduced large-scale datasets such as HSpam14 [4] and conducted in-depth analyses of hashtag-oriented spam behavior across millions of tweets [3]. Their findings demonstrated that hashtags are frequently misused for unsolicited advertising, misleading links, and low-quality promotional content. Subsequent research examined the impact of spam on hashtag recommendation systems, showing that unsolicited promotional tweets significantly degrade recommendation quality and overall user experience [12].

Another line of research focuses on detecting large-scale promotional campaigns at the platform level. Zhang et al. proposed a behavioral detection framework that classifies accounts into normal, spam, and promotional campaigns by analyzing posting patterns and URL-sharing behaviors [5]. While these studies confirm the prevalence and sophistication of hashtag-oriented promotional activities, they primarily operate at the account or network level. However, such approaches do not explicitly address the identification of promotional content at the individual tweet level within specific hashtag streams. In contrast, the present work focuses on fine-grained tweet-level analysis, aiming to identify promotional tweets and assess their impact on the quality of hashtag-based discussions.

B. Topical Analysis and Relevance in Twitter

Analyzing topical content in short texts such as tweets is inherently challenging due to sparsity, noise, and limited context. Early studies highlighted the limitations of traditional topic modeling techniques, such as Latent Dirichlet Allocation (LDA), when applied to Twitter data and proposed adaptations for microblog environments [13]. Hong and Davison examined topic modeling in Twitter and demonstrated the importance of adapting topic-model training to short microblog messages [13].

Hashtag structures have been used to enhance topical interpretation, including graph-based hashtag analysis [14]. More recent work has established transformer-based tweet classification as a strong approach across heterogeneous social-media tasks [6]. While these methods are effective in discovering latent thematic structures, they primarily focus on topic extraction rather than evaluating whether individual tweets semantically align with the intended meaning of a specific hashtag. In contrast, our work adopts a relevance-based perspective, where predefined hashtag topic descriptions and embedding-based similarity measures are used to distinguish on-topic and off-topic tweets within hashtag streams.

C. NLP and Machine Learning for Arabic Twitter Analysis

Arabic Twitter content presents additional challenges due to the language's rich morphology, dialectal variation, and

informal writing styles. Traditional lexical representations such as TF–IDF remain useful baselines for social-media classification, while standardized tweet-classification benchmarks have increasingly emphasized transformer-based representations [6].

The introduction of transformer-based language models pretrained on Arabic corpora, such as ARBERT, MARBERT, and AraBERT, has significantly improved performance across a variety of Arabic Twitter analysis tasks [9], [10]. These models have demonstrated strong effectiveness in areas such as sentiment analysis, spam detection, sarcasm recognition, and offensive language classification [6], [11]. Despite these advances, relatively limited work has explored hashtag-centered analytical frameworks that jointly assess topical relevance and promotional content at the tweet level within Arabic hashtag streams.

D. Summary

Recent work on the evolution of cloud-to-edge digital applications further illustrates the broader movement toward intelligent, data-driven services and machine-learning-enabled analytics [15]. Overall, prior research has made substantial contributions to spam detection, topical analysis, and Arabic NLP. However, most existing studies either focus on discovering latent topics or detecting spam and promotional campaigns at the account or network level. Few works address the combined problem of assessing tweet-level topical relevance and detecting promotional content within individual hashtags, particularly in Arabic contexts. This study addresses this gap by proposing a hybrid, hashtag-centered NLP framework that integrates embedding-based topic relevance detection with supervised promotional-content classification to improve the reliability and interpretability of hashtag-based Twitter analysis.

III. Materials and Methods

This section describes the dataset construction process, data collection strategy, preprocessing pipeline for Arabic tweets, and the feature representations used in this study. The objective is to enable tweet-level analysis of hashtag streams for 1) topic relevance assessment and 2) advertisement detection.

A. Dataset Collection

Arabic tweets were collected through the official X (formerly Twitter) API while the selected Arabic hashtags were active. The final archived research files supplied for this study contain 9,671 Arabic-language records spanning 9 October–15 December 2025 (UTC). The six streams cover technology, social issues, religious discussion, sports, and public-event topics. Their archived sizes are 501 tweets for #War_on_Drugs, 309 for #Data_Analysis, 2,162 for #Artificial_Intelligence, 1,908 for #World_Cup, 1,772 for #Friday, and 3,019 for #Study_Suspension. All records in these six files are marked with the Arabic language code extttar.

The archived files preserve tweet identifiers, timestamps, conversation identifiers, account identifiers, public profile metadata, and engagement fields. Exact duplicate checks on the supplied files identified 637 repeated TweetID occurrences and 817 repeated text occurrences across the combined archive; these fields make duplicate-aware filtering and user-aware evaluation possible. Each record includes core fields provided by the API:

- Tweet and conversation identifiers,
- Tweet text content and timestamp,
- Public account/profile metadata and engagement fields when available.

Only publicly available posts were used. The archived research files retain the fields required for reproducibility, including TweetID, ConversationID, text, date, language, engagement counts, username, public profile location, verification status, biography, and follower/following statistics. Personally identifying account fields are not required by the proposed text-classification models and should not be redistributed in a public derived dataset.

The archive contains 4,433 distinct usernames; TweetID, username, and text therefore provide explicit keys for duplicate-aware and user-disjoint validation.

Fig. 1 presents English translations of the analyzed hashtags, and Fig. 2 shows the distribution of tweets across hashtags.

Hashtag in English	Hashtag in Arabic
#War_on_Drugs	#الحرب_على_المخدرات
#Data_Analysis	#تحليل_البيانات
#Artificial_Intelligence	#الذكاء_الاصطناعي
#World_Cup	#كأس_العالم
#Class_Suspension	#تعليق_الدراسة
#Friday	#يوم_الجمعة

Fig. 1. English translation of the Arabic hashtags analyzed in this study.

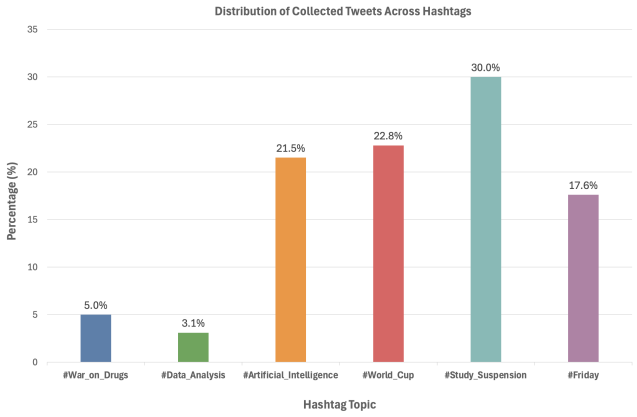


Fig. 2. Percentage distribution of tweets across the selected hashtags.

B. Data Preprocessing

Tweets are noisy and informal by nature (e.g., URLs, mentions, emojis, repeated characters). Arabic content further increases complexity due to rich morphology, dialectal variation, and non-standard spelling in social media. A multi-step preprocessing pipeline was applied to clean and normalize tweet text:

- Text cleaning: Removal of URLs, user mentions (@), emojis, hashtag symbols, and irrelevant special characters while preserving core semantic content.
- Arabic normalization: Unification of letter variants, removal of diacritics and tatweel, reduction of elongated/repeated characters, correction of common orthographic inconsistencies, and normalization of punctuation.
- Tokenization: Segmentation into tokens using Arabic-appropriate rule-based tokenization.
- Stop-word removal: Removal of frequent Arabic stop words using standard lists, supplemented with social-media filler terms when needed.

To quantify the effect of preprocessing, Fig. 3 compares average tweet length before vs. after preprocessing, showing a clear reduction due to removal of non-informative noise.

Hashtags	Avg. words before preprocessing	Avg. words after preprocessing	Noise reduction in text
#War_on_Drugs	18.58	11.12	%40.12
#Data_Analysis	30.93	16.33	%47.20
#Artificial_Intelligence	28.65	18.41	%35.75
#World_Cup	22.02	12.44	%43.52
#Class_Suspension	24.73	16.72	%32.40
#Friday	27.08	19.77	%27.02

Fig. 3. Average tweet length before vs. after preprocessing.

C. Exploratory Metadata Analysis

Exploratory analysis was performed on selected metadata fields to contextualize hashtag dynamics. User-provided location information was examined to capture broad participation patterns across regions, noting that many users omit location and are therefore treated as unknown. Additionally, user activity was analyzed by computing tweet frequency per user within each hashtag to assess whether content was concentrated among a small number of accounts.

During Arabic text visualization (e.g., word-frequency plots), standard plotting libraries may render Arabic characters incorrectly (disconnected or reversed) due to left-to-right layout. When such visualizations were used, Arabic reshaping was applied before plotting to ensure correct right-to-left rendering and improved readability.

D. Feature Representation

Two complementary feature representations were employed, aligned with the framework's two tasks:

- TF–IDF vectors: Extracted from the preprocessed tweet text (`Clean Text`) and used for supervised advertisement classification. This representation captures lexical cues and recurring promotional patterns.
- Transformer-based contextual embeddings: Contextual representations generated using AraBERT (`aubmindlab/bert-base-arabertv02`) and used to compute cosine similarity between tweet representations and predefined hashtag topic descriptions for relevance assessment. The archived CSV files do not contain cached embedding tensors; unverifiable implementation details are therefore not asserted.

E. Topic Relevance Detection Module

Topic relevance detection evaluates whether a tweet aligns semantically with the intended meaning of its hashtag. For each hashtag, a concise topic description was manually defined to represent its semantic scope. AraBERT embeddings were computed for tweets and topic descriptions, and cosine similarity was used to quantify semantic relatedness. A fixed decision threshold, established during the original experimental analysis, classified tweets as *on-topic* or *off-topic*. The archived CSV files do not store similarity scores or threshold values; no unverified threshold is therefore reported.

The semantic relevance module was evaluated independently from advertisement classification: its on-topic/off-topic decision was based only on the cosine-similarity rule. Advertisement predictions were incorporated only in the final fused stream label, where promotional tweets were excluded from the organic on-topic subset. This separation preserves independent assessment of semantic relevance while retaining the practical filtering objective of the hybrid framework.

F. Advertisement Classification

Advertisement detection was formulated as supervised binary classification, independent of topic relevance detection. TF–IDF features were extracted from `Clean Text`, and a binary label `Has_Ad` was used as the target variable (`1` for promotional content; `0` otherwise).

Prior to training, the dataset was split into training and testing subsets using stratified sampling to preserve label proportions under class imbalance. Five classifiers were evaluated:

- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- Naive Bayes
- Stochastic Gradient Descent (SGD) Classifier

This module outputs `Has_Ad` predictions that are later integrated into the final relevance labeling (Section IV).

G. Evaluation Metrics

Advertisement classification was evaluated using Accuracy, Precision, Recall, and F1-score:

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN}, \\ \text{Precision} &= \frac{TP}{TP+FP}, \\ \text{Recall} &= \frac{TP}{TP+FN}. \end{aligned} \tag{1}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2}$$

where, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

Topic relevance detection was assessed through the observed off-topic proportions and comparison with manual off-topic estimates reported in Section V. These manual estimates are used as a descriptive comparison rather than as a held-out accuracy benchmark; consequently, no unsupported accuracy statistic is reported for the relevance module.

IV. Methodology

This section presents the proposed hybrid NLP framework for analyzing hashtag-based streams on X. The framework combines semantic topic relevance detection using transformer embeddings with supervised advertisement classification using TF–IDF features. Fig. 4 illustrates the overall pipeline.

A. Overview of the Proposed Framework

The processing workflow consists of four stages:

- Hashtag-based collection and preprocessing: Tweets are collected per hashtag and standardized using the preprocessing pipeline described in Section III.
- Parallel feature generation: AraBERT embeddings are generated for topic relevance detection, and TF–IDF vectors are generated for advertisement classification.
- Topic relevance detection: Tweet embeddings are compared with the corresponding hashtag topic description embedding using cosine similarity and a threshold rule to classify tweets as on-topic or off-topic.
- Advertisement classification: A supervised model predicts whether a tweet is promotional using TF–IDF features and the `Has_Ad` label.

B. System Architecture

The pipeline branches into two parallel analysis paths after feature representation: 1) topic relevance detection based on semantic similarity and 2) advertisement classification based on supervised learning. The two branches are evaluated separately. Their outputs are combined only at the final filtering stage, where predicted advertisements are excluded from the organic on-topic subset.

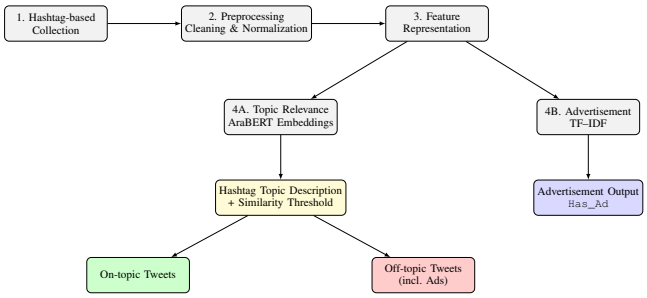


Fig. 4. Architecture of the proposed hybrid NLP framework for embedding-based topic relevance detection and supervised advertisement classification within hashtag-based streams on X.

C. Topic Relevance Detection

For each hashtag, a topic description is manually defined to represent the intended semantic scope. AraBERT embeddings are computed for tweets and topic descriptions, and cosine similarity is used to measure semantic alignment. Tweets are labeled as on-topic when their similarity exceeds the decision threshold and off-topic otherwise. This semantic decision is retained independently for evaluation. In the final fused output only, predicted advertisements are excluded from the organic on-topic subset so that advertisement-classifier errors do not alter the reported semantic-module decision.

D. Advertisement Classification

Advertisement detection is implemented as a binary decision module that flags whether a tweet contains promotional intent. Given a cleaned tweet, the classifier outputs a label $\texttt{Has\_Ad} \in \{0, 1\}$, where $\texttt{Has\_Ad} = 1$ denotes advertisement content. This output provides a direct estimate of promotional intrusion within each hashtag stream and is combined with the semantic label only in the final filtering stage. The advertisement prediction therefore does not overwrite the independently reported semantic relevance decision.

V. Results and Discussion

This section reports the experimental results of the proposed hybrid framework, covering:

- Supervised advertisement detection using TF–IDF features and classical machine learning classifiers and
- Embedding-based topic relevance detection using AraBERT similarity scores.

In addition to per-hashtag model tables, we provide cross-hashtag summary visualizations and aggregated comparisons to highlight robustness, variability across topics, and practical implications for hashtag stream credibility.

A. Advertisement Classification Results

Advertisement detection was formulated as a binary classification task ($\texttt{Has\_Ad} \in \{0, 1\}$), using TF–IDF features extracted from the cleaned tweet text. Performance was evaluated using Accuracy, Precision, Recall, and F1-score. Because promotional tweets are typically the minority class in several hashtags, Precision and Recall provide a more faithful assessment than Accuracy alone.

1) Per-hashtag performance: Tables I–VI summarize model performance for each hashtag. Overall, we observe that model behavior varies substantially across hashtags, reflecting differences in topic specificity, lexical patterns of promotions, and class imbalance.

TABLE I. Advertisement Classification for #War_on_Drugs

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.940	0.142	1.00	0.25
SVM	0.950	0.166	1.00	0.28
Random Forest	0.950	0.166	1.00	0.28
SGDClassifier	0.950	0.166	1.00	0.28

Interpretation. The very high Recall indicates that most advertisements are detected, but low Precision implies more false positives. This is consistent with strong imbalance or overlap between promotional keywords and organic discussion terms in this stream.

TABLE II. ADVERTISEMENT CLASSIFICATION FOR #FRIDAY

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.960	0.444	0.666	0.533
SVM	0.932	0.285	0.666	0.400
Random Forest	0.923	0.222	0.500	0.307
Naive Bayes	0.977	1.000	0.333	0.500

Interpretation. Naive Bayes achieves extremely high Precision but low Recall, meaning it flags ads only when evidence is very strong, missing many subtle advertisements. Logistic Regression provides a better Precision–Recall tradeoff for this hashtag.

TABLE III. ADVERTISEMENT CLASSIFICATION FOR #DATA_ANALYSIS

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.887	0.692	0.750	0.720
SVM	0.903	0.750	0.750	0.750
Random Forest	0.935	1.000	0.667	0.800
Naive Bayes	0.903	1.000	0.500	0.667

Interpretation. Random Forest achieves the best F1-score, suggesting it captures non-linear combinations of TF–IDF signals that are characteristic of promotional tweets in a technical hashtag.

TABLE IV. ADVERTISEMENT CLASSIFICATION FOR #ARTIFICIAL_INTELLIGENCE

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.983	1.000	0.788	0.881
SVM	0.991	1.000	0.879	0.935
Random Forest	0.983	1.000	0.788	0.881
Naive Bayes	0.942	1.000	0.273	0.418

Interpretation. This hashtag yields consistently high Precision across models, indicating that advertisements contain strong lexical cues. SVM produces the highest F1-score by maintaining high Recall while preserving perfect Precision.

TABLE V. ADVERTISEMENT CLASSIFICATION FOR #WORLD_CUP

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.976	0.500	0.455	0.476
SVM	0.976	0.500	0.455	0.476
Random Forest	0.985	1.000	0.364	0.533
Naive Bayes	0.976	0.500	0.364	0.476

Interpretation. General-event hashtags often contain diverse language and high lexical overlap between organic posts and promotions, reducing separability. Random Forest achieves the best F1 primarily via higher Precision.

TABLE VI. ADVERTISEMENT CLASSIFICATION FOR #CLASS_SUSPENSION

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.900	0.770	0.750	0.760
SVM	0.890	0.750	0.750	0.750
Random Forest	0.920	0.920	0.710	0.800
Naive Bayes	0.920	0.940	0.660	0.770

Interpretation. Random Forest yields the strongest balance between Precision and Recall, suggesting that promotional tweets in this stream are detectable but require modeling interactions between multiple lexical indicators.

2) Cross-hashtag summary (best model + robustness): To summarize performance trends, Table VII reports the best-performing classifier per hashtag based on F1-score. Fig. 5 provides a compact visualization of the F1 variation across models and hashtags, and Fig. 6 highlights the Precision–Recall tradeoff, which is especially important under class imbalance.

TABLE VII. BEST ADVERTISEMENT CLASSIFIER PER HASHTAG BASED ON F1-SCORE

Hashtag	Best Model (F1)	F1-score
#War_on_Drugs	SVM / RF / SGD (tie)	0.28
#Friday	Logistic Regression	0.533
#Data_Analysis	Random Forest	0.800
#Artificial_Intelligence	SVM	0.935
#World_Cup	Random Forest	0.533
#Class_Suspension	Random Forest	0.800

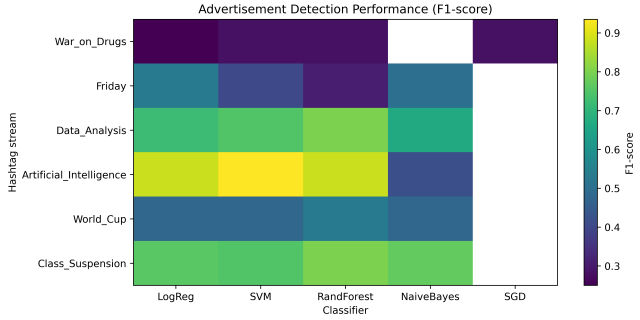


Fig. 5. F1-score heatmap for advertisement detection across hashtags and classifiers. The figure highlights that model effectiveness is hashtag-dependent, motivating per-hashtag model selection rather than a single universal classifier.

B. Topic Relevance Detection Results

Fig. 7 summarizes the best advertisement-detection F1-score for each hashtag and highlights the strong dependence of classifier performance on hashtag context.

Topic relevance detection was performed using AraBERT embeddings and cosine similarity between each tweet and a predefined hashtag topic description. The percentages reported here are produced by the semantic relevance branch before advertisement-based fusion. Table VIII compares manual off-topic estimation versus embedding-based detection.

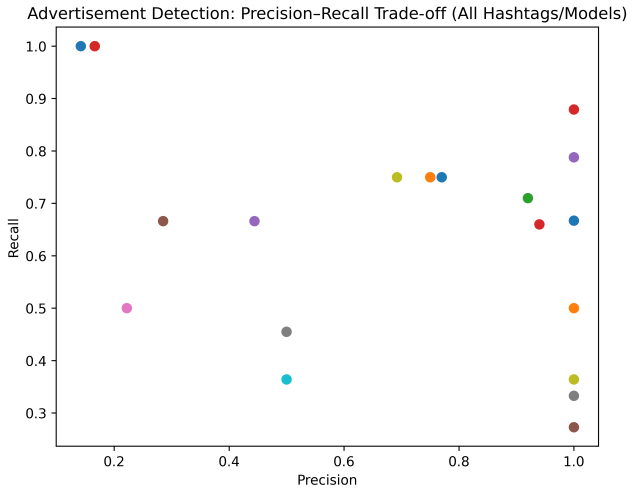


Fig. 6. Precision–Recall tradeoff across hashtags and models for advertisement detection. Models with high Recall but low Precision tend to over-flag promotions, while high Precision with low Recall indicates missing subtle advertisements.

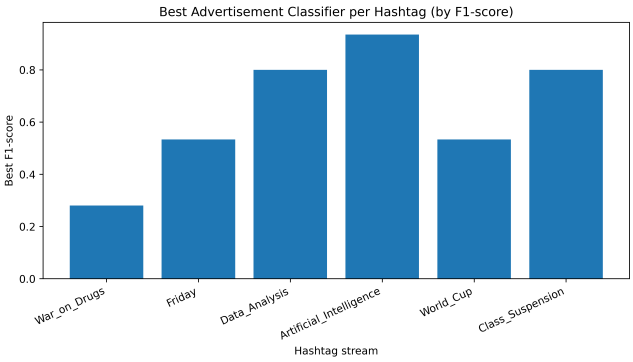


Fig. 7. Best F1-score achieved per hashtag for advertisement detection. The variability confirms that topic specificity and class balance strongly influence separability.

TABLE VIII. OFF-TOPIC DETECTION RESULTS (MANUAL VS. EMBEDDING-BASED MODEL)

Hashtag	Total Tweets	Off-topic (Manual %)	Off-topic (Model %)
#War_on_Drugs	501	2.60	3.40
#Data_Analysis	309	46.28	55.00
#World_Cup	2292	3.18	60.00
#Study_Suspension	3019	59.88	91.40
#Artificial_Intelligence	2162	9.34	89.00
#Friday	1772	10.80	67.00

Interpretation. The embedding-based model often reports higher off-topic proportions than manual inspection, especially for highly active hashtags. This suggests that semantic similarity filtering captures subtle drift and unrelated conversational branches that may not be counted as spam/ads during manual checks. At the same time, the gap indicates that similarity thresholds can be conservative, which motivates future calibration using larger annotated relevance datasets.

C. Comparative Discussion and Key Findings

Fig. 8 and 9 complement Table VIII by visualizing the manual–model off-topic percentages and their differences across hashtags.

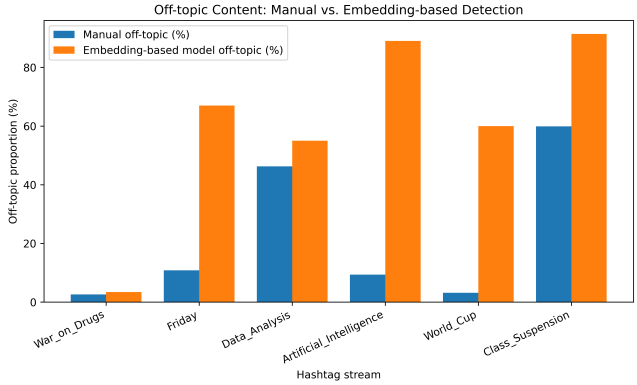


Fig. 8. Manual vs. embedding-based off-topic percentage across hashtags. The embedding-based method generally applies a stricter semantic criterion, yielding higher off-topic rates in high-activity streams.

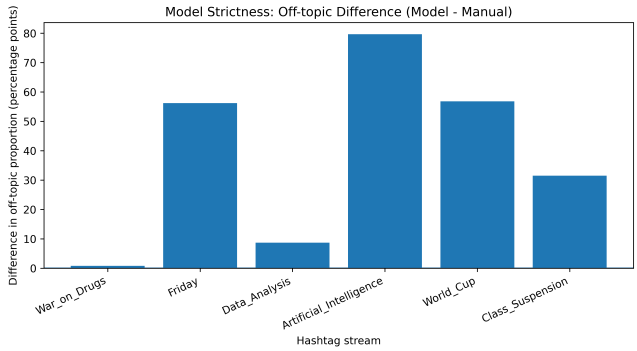


Fig. 9. Difference between embedding-based and manual off-topic estimates (Model% – Manual%). Large positive gaps indicate streams where semantic drift is strong or where manual inspection underestimates off-topic diversity.

Across tasks, the results support three consistent observations:

1) Hashtag dependency is substantial: Advertisement detection performance varies widely between technical hashtags (more consistent promotional language) and broad event hashtags (higher linguistic diversity and overlap). This justifies evaluating models per hashtag rather than reporting a single global score.

2) Precision–recall tradeoffs reveal practical behavior under imbalance: Several models achieve high Accuracy while yielding weak Precision, meaning they may over-flag ads when the positive class is rare. Therefore, Precision, Recall, and F1-score provide more informative signals for model selection.

3) Embedding-based relevance detection is sensitive but can be conservative: The topic relevance module captures semantic divergence beyond explicit advertisements/spam, improving stream credibility filtering. However, the model may label a larger fraction of tweets as off-topic in high-activity

streams, indicating that threshold calibration and richer ground truth labels would further improve reliability.

In summary, combining TF–IDF-based supervised detection for promotional intent with AraBERT-based semantic relevance filtering provides a practical and interpretable framework for improving the trustworthiness of trending hashtag streams, especially in Arabic contexts.

VI. Conclusion

This study presented a hybrid NLP framework for improving the reliability of trending hashtag streams on Twitter/X, with a focus on Arabic-language content. The proposed approach integrates two complementary components: 1) embedding-based topic relevance detection, which measures semantic alignment between tweets and the intended hashtag meaning using AraBERT representations, and 2) supervised advertisement detection using TF–IDF features with classical machine learning classifiers. By separating *semantic relevance* from *promotional intent*, the framework provides clearer interpretability of hashtag dynamics and supports fine-grained filtering of noisy content while preserving meaningful organic discussion.

Experimental results across multiple trending hashtags indicate that advertisement detection is generally stronger in domain-specific hashtags (e.g., technology-oriented streams), where promotional language is more consistent and lexically separable, while broader event-oriented hashtags exhibit greater linguistic diversity and lexical overlap between organic and promotional tweets, leading to more variable performance. In parallel, the topic relevance module often reports higher off-topic proportions than manual inspection in highly active hashtags, suggesting that contextual embeddings can capture subtle semantic drift beyond surface keyword matching. Manual off-topic estimates were retained as a descriptive comparison with the embedding-based outputs rather than being presented as an independent accuracy benchmark.

Overall, the findings confirm that combining contextual semantic representations with lightweight supervised classifiers is an effective and scalable strategy for analyzing Arabic hashtag streams and mitigating promotional intrusion. Future improvements will focus on expanding and refining manual annotation to enable stronger statistical evaluation, replacing fixed similarity thresholds with learning-based relevance models, and exploring transformer-based or hybrid classifiers for advertisement detection to better capture implicit promotional intent. Additional extensions include imbalance-aware calibration to reduce false positives when ads are rare, and incorporating temporal and metadata signals (e.g., activity bursts, repeated posting patterns) to better distinguish coordinated campaigns from organic engagement, as well as validating the framework in multilingual settings and across other social media platforms.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFRGS-2026".

References

[1] P. Gundecha and H. Liu, "Mining social media: A brief introduction," in *2012 TutORials in Operations Research: New Directions in Informatics, Optimization, Logistics, and Production*, INFORMS, 2012, pp. 1–17, doi: 10.1287/educ.1120.0105.

[2] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*. Cambridge, U.K.: Cambridge University Press, 2014, doi: 10.1017/CBO9781139088510.

[3] S. Sedhai and A. Sun, "An analysis of 14 million tweets on hashtag-oriented spamming," *Journal of the Association for Information Science and Technology*, vol. 68, no. 7, pp. 1638–1651, 2017, doi: 10.1002/asi.23836.

[4] S. Sedhai and A. Sun, "HSpam14: A collection of 14 million tweets for hashtag-oriented spam research," in *Proc. 38th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2015, pp. 223–232, doi: 10.1145/2766462.2767701.

[5] X. Zhang, Z. Li, S. Zhu, and W. Liang, "Detecting spam and promoting campaigns in Twitter," *ACM Transactions on the Web*, vol. 10, no. 1, Art. 4, pp. 1–28, 2016, doi: 10.1145/2846102.

[6] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1644–1650, doi: 10.18653/v1/2020.findings-emnlp.148.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing*, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[9] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proc. 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, 2020, pp. 9–15, doi: 10.48550/arXiv.2003.00104.

[10] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," in *Proc. 59th Annual Meeting of the Association for Computational Linguistics and 11th Int. Joint Conf. Natural Language Processing*, 2021, pp. 7088–7105, doi: 10.18653/v1/2021.acl-long.551.

[11] A. Wadhawan, "AraBERT and Farasa segmentation based approach for sarcasm and sentiment detection in Arabic tweets," in *Proc. Sixth Arabic Natural Language Processing Workshop*, 2021, pp. 395–400, doi: 10.48550/arXiv.2103.01679.

[12] S. Sedhai and A. Sun, "Effect of spam on hashtag recommendation for tweets," in *Proc. 25th Int. Conf. World Wide Web Companion*, 2016, doi: 10.1145/2872518.2889404.

[13] L. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," in *Proc. First Workshop on Social Media Analytics*, 2010, pp. 80–88, doi: 10.1145/1964858.1964870.

[14] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in Twitter: A graph-based hashtag sentiment classification approach," in *Proc. 20th ACM Int. Conf. Information and Knowledge Management*, 2011, pp. 1031–1040, doi: 10.1145/2063576.2063726.

[15] S. M. Altowaijri and M. Ayari, "The synergistic impact of 5G on cloud-to-edge computing and the evolution of digital applications," *Mathematics*, vol. 13, no. 16, Art. 2634, 2025, doi: 10.3390/math13162634.

Enhanced Harris Hawks Optimization Algorithm with Integrated Tabu Search for 3D Bin Packing Problems: Application in Logistics Industries

Yasser M. Ayid[1]*, Mohamed Meselhy Eltoukhy[2], Alaa Mokhtar[3], Rabie Mosaad[4]
Department of Computer Science and Artificial Intelligence-College of Computer Science and Engineering, University of Jeddah, 23890 Jeddah, Saudi Arabia[1]
Department of Information Technology-College of Computing and Information Technology at Khulais, University of Jeddah, 23890 Jeddah, Saudi Arabia[2]
Canadian International College, School of Business Technology, Zayed Campus, 6th of October 12585, Egypt[3]
Department of Operations Research and Management-Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt[4]

***Abstract*—The Harris Hawks Optimization (HHO) algorithm has proven to be an effective metaheuristic tool for addressing complex optimization challenges. This study outlines significant modifications and enhancements proposed for the standard HHO algorithm to improve its performance on complex three-dimensional bin packing problems (3D-BPP). The proposed modifications introduce a hybrid metaheuristic framework that synergistically merges the global search capabilities of an enhanced HHO with the local intensification strength of Tabu Search (TS), along with several additional algorithmic improvements. In this study, we present an enhanced Harris Hawks Optimization algorithm integrated with Tabu Search for 3D bin packing problems (EHHITS). This problem is particularly significant in logistics and warehousing, where improving space utilization directly contributes to cost reduction. The primary objective is to maximize the occupied volume of the container by efficiently packing smaller rectangular items (boxes) without overlaps. The proposed EHHITS algorithm employs an inner heuristic, a heuristic local search, to construct an optimized sequence of boxes, which is then used to determine their placement within the container. To evaluate the proposed algorithm, we benchmarked its performance against standard test cases from the literature. The experimental results demonstrate its effectiveness and validity.**

***Keywords*—Logistics; container loading problems; harris hawks optimization; tabu search; wall-building**

I. Introduction

The 3D bin packing problem is a critical challenge in logistics and transportation. It focuses on efficiently arranging a collection of items within a container to achieve maximum utilization of its available volume. This optimization is essential for companies seeking to minimize shipping costs and enhance customer satisfaction in a globalized market. The 3D bin packing problem becomes increasingly difficult to solve optimally as the number of items grows [1]. The goal is to pack a set of n items, each with defined dimensions, into a single container of known size. The core constraints require that no item exceeds the container's dimensions and that items do not overlap in space. The objective is to achieve the highest possible container utilization rate [2], [3].

*Corresponding author.

HHO is a population-based metaheuristic that emulates the collective predatory behavior of Harris hawks. HHO outperforms many other algorithms due to its inherent strengths [4]. A key advantage is its strong balance between global exploration and local exploitation, exhibiting high optimization efficiency in both phases. This robust search strategy gives HHO strong solution-finding capability, making it particularly effective for challenging NP-hard problems [5]. Moreover, the algorithm dynamically adjusts its search behavior throughout the optimization process [6]. As a result, these superior characteristics have led to the widespread and recent adoption of HHO and its variants in solving numerous real-world problems.

In addition to its strong theoretical foundation, the versatility of the HHO algorithm is demonstrated through its successful application across a wide range of fields. Recent studies highlight its effectiveness in critical domains such as estimating parameters of single-diode and three-diode photovoltaic models [7], [8]. Beyond renewable energy, HHO has been employed in various image segmentation tasks including digital mammogram and satellite image segmentation [9], as well as in fault detection for anti-friction bearings [10]. Recent publications further confirm the effectiveness and continued relevance of the HHO algorithm.

The motivations of this study are as follows:

- To develop an enhanced HHO algorithm integrated with Tabu Search for solving the 3D bin packing problem.
- To introduce and demonstrate an enhanced version of the algorithm, namely EHHITS, through the integration of effective search mechanisms and improved solution-generation strategies, leading to robust and competitive performance on a comprehensive set of three-dimensional packing instances.

This research presents an improved variant of the HHO algorithm integrated with Tabu Search. The proposed algorithm is evaluated against well-established approaches in the literature within the context of 3D bin packing. The structure

of this study is organized to thoroughly address the 3D bin packing problem using an HHO-based methodology. The study begins with a comprehensive literature review in Section II, evaluating existing heuristic and metaheuristic algorithms for 3D bin packing and comparing them with alternative solution techniques. Section III explains the HHO algorithm. Section IV introduces the core contribution of this work. Finally, Section V presents a detailed analysis of the computational results, followed by concluding remarks that summarize the findings and validate the effectiveness of the proposed approach.

II. Literature Review

The Harris Hawks Optimization algorithm has attracted considerable interest in the academic community since its conception, thanks to its structural flexibility and strong performance in tackling complicated optimization problems. As shown in Table I, the literature shows a strong trend toward hybridizing and improving the underlying HHO architecture to address domain-specific problems.

Researchers addressing the 3D bin packing problem primarily rely on three categories of optimization algorithms [34]. Although exact methods guarantee optimal solutions, they quickly become computationally infeasible as the number of boxes increases. In contrast, metaheuristic algorithms cannot guarantee a global optimum, but they can produce near-optimal and practically acceptable solutions within reasonable computation time [35]. For this reason, Metaheuristic Algorithms (MHAs) have become the preferred choice among researchers tackling 3D bin packing. There are three main strategies for applying MHAs to solve 3D bin packing:

1) Direct application: In this basic approach, MHAs are applied "as-is" without modification. It is the simplest strategy and often serves as a baseline for performance comparison.

2) Modification: This strategy involves tailoring an existing MHA to better address the specific characteristics of the 3D bin packing problem. Adjustments may include parameter tuning or incorporating new search mechanisms to improve efficiency and solution quality.

3) Hybridization: The most complex approach, hybridization combines two or more MHAs or integrates an MHA with a different type of algorithm. The objective is to exploit the complementary strengths of each component to achieve superior performance.

The three-dimensional container loading problem (3D-CLP) deals with placing rectangular cargo items inside a container while making the best possible use of the available space. The objective is to maximize the proportion of the container volume occupied by the loaded items, as expressed in Eq. (1).

$$\max f = \frac{\sum_{i=1}^{n} v_i \cdot k_i}{L \times W \times H} \tag{1}$$

where, n denotes the number of cargo types, v_i represents the volume of each cargo item of type i, and k_i denotes the number of items of type i loaded into the container. The parameters L, W, and H represent the length, width, and height of the container, respectively.

TABLE I. Several Key Studies Concerning HHO

Approach	Application area	Ref.
HHO-Multi-Layer Perceptron	Two-layer foundation soils	[11]
HHO and Simulated Annealing	Feature selection	[12]
HHO	Design of microchannel heat sinks	[13]
Vibrational HHO	Fault diagnosis of rolling bearing	[14]
Enhanced HHO	Breast Cancer Detection	[15]
Whippy HHO	PV solar cells	[16]
Artificial Neural Network	Stability of the soil slopes	[17]
HHO Microgrid	Design optimized	[18]
Sine-Cosine HHO	Feature selection CEC2017	[19]
Deep Neural Network - HHO	Evaluating slope stability	[20]
Improved Harris Hawks Optimizer	Global optimization	[21]
Improved HHO	Collaborative optimization	[22]
Multi-Objective Binary HHO	Gene selection	[23]
Deep Learning - HHO	Skin lesion analysis	[24]
HHO with Chaotic Sequence	Engineering problems	[25]
Hybrid HHO with Chaotic Maps	Mechanical design problems	[26]
Improved HHO	Path planning for mobile robots	[27]
HHO	Machine learning	[28]
Enhanced HHO	Task scheduling in cloud computing	[29]
HHO	Power systems and energy management	[30]
Improved HHO	Optimizing deep learning models	[31]
Enhancing HHO	Global optimization problems	[32]
Hybrid Aquila optimizer HHO	CNN hyper parameter	[33]

In practical container loading, several constraints must be considered in addition to the dimensional and volume limitations. These include the available space for the cargo, cargo stability, the center of gravity of the container, and the prevention of overlap between cargo items. Such constraints can have a direct impact on the quality of the obtained packing solution. Therefore, the characteristics of the specific packing problem should be carefully considered. Based on the test dataset, the following constraints are considered for the 3D-CLP:

- The cargo items and the container are rectangular or square in shape.
- The cargo items are placed within the positive coordinate directions of the container.
- The available cargo can be loaded without considering crushing of the cargo items in the lowest layer.
- The cargo items are placed parallel to the axes of the container, and overlapping between cargo items is not allowed.
- The dimensions of each cargo item must be smaller than the corresponding dimensions of the container.

These assumptions provide a simplified yet suitable representation of the 3D-CLP considered in this study. The complexity of the 3D bin packing problem can vary significantly depending on the number and characteristics of the items to be loaded. These variations range from the homogeneous case as shown in Fig. 1, where all items share the same orientation and dimensions, to the strongly heterogeneous case, which involves a wide variety of item sizes. The problem becomes more challenging to solve as item diversity increases [36].

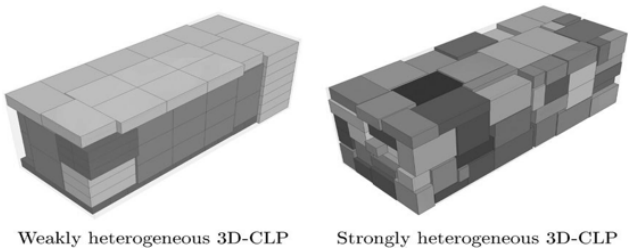


Fig. 1. Weakly and strongly heterogeneous 3D bin packing [37].

A. Packing Heuristics

Fanslau and Bortfeldt [38] classified the wide range of approaches for solving the 3D bin packing problem based on both the packing heuristic and the type of solution method employed. Packing heuristics define the specific strategies used to place items within the container, these strategies are categorized into several groups.

1) Wall-building: Bortfeldt and Gehring [2] introduced a heuristic wall-building within a hybrid Genetic Algorithm (GA) designed for packing a single container with boxes of varying sizes. In their approach, the stowage plan is represented using a complex, problem-specific data structure. This structure enables the construction of loading plans of several vertical layers, each containing multiple boxes. GA then optimizes these layer-based arrangements to produce highly efficient packing solutions.

2) Stack-building: Bischoff and Ratcliff [39] identified significant gaps in the current theoretical framework on Container Loading Problems. They argued that current methods are inadequate for many practical situations and that fundamentally new approaches are required. Moreover, they emphasized that the development of such methods should include extensive experimentation to assess their strengths and weaknesses.

3) Horizontal layer-building: Terno et al. [40] implemented their approach as a foundation for further research, particularly in the context of the Multi-Product Lot-Sizing Problem (MPLP). They also considered a more general scenario where the demand for a certain type of item is not a fixed quantity but is defined by a lower and upper bound. This adds another layer of complexity to the problem, making it more representative of real-world logistics.

4) Block-building: Eley [41] developed a homogeneous block arrangement to solve single-container problems. The use of these blocks offers several advantages for practical applications. When tested against standard benchmarks, it performed well in terms of volume utilization in heterogeneous problems. This demonstrates the effectiveness of the block-building approach in complex real-world scenarios.

5) Guillotine cutting: Morabito [42] addressed two-dimensional unconstrained complex cutting problems.Where a rectangular plate is divided into smaller rectangles using cuts perpendicular to a single edge. Unlike traditional guillotine methods, this approach allows for more complex cutting patterns. Morabito's method uses a graph-search technique to find an efficient cutting plan, which can be applied to container loading as a way to organize and place items.

As illustrated in Fig. 2 the evolution of container loading and packing methodologies demonstrates a clear progression from basic geometric partitioning to sophisticated heuristic optimization. Early techniques, such as the guillotine-cutting approach, relied on a structured slicing tree to divide space, whereas subsequent developments introduced more practical organizational logic through vertical stack-building and horizontal layer-building. As the field matured, methodologies shifted toward vertical wall-building and complex block-building strategies, which utilize tree-search algorithms to maximize spatial density. This transition reflects a shift in industrial research from solving simple rectangular constraints toward achieving high-volume utilization and structural stability in complex logistical environments.

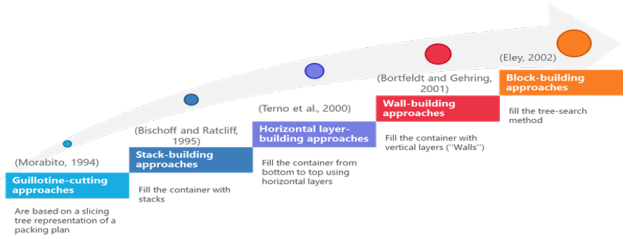


Fig. 2. 3D bin packing according to packing heuristic.

B. Metaheuristics Search Strategies

Metaheuristics search strategies have become the preferred approach for solving complex optimization problems over the past decade. Akl et al. [43] illustrate the workings of these population-driven techniques as in Fig. 3. The process begins with a diverse initial population that undergoes continuous refinement through fitness evaluations. At the heart of the strategy is a dynamic balance between exploration seeking new possibilities and exploitation focusing on the most promising solutions. This cycle iterates intelligently, progressively narrowing down options until the stopping criteria are fulfilled and the optimal solution is achieved. A notable example is the Tabu Search Approach (TSA), which has been effectively applied to the Container Loading Problem. Bortfeldt and Gehring [2] developed a parallel Tabu Search algorithm for solving the single-container loading problem, where multiple cooperating search threads are used to enhance computational efficiency. Their extensive comparative experiments demonstrated that even the sequential version of TSA can achieve high container volume utilization, underscoring the strength of this metaheuristic method in tackling difficult combinatorial optimization tasks.

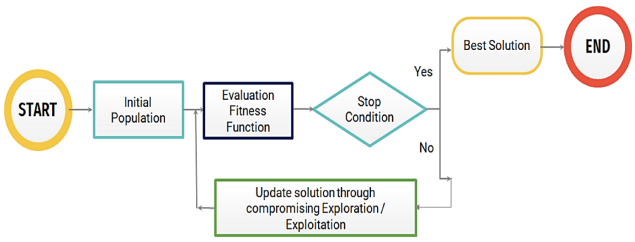


Fig. 3. Conceptual flowchart of population-driven metaheuristic optimization techniques [43].

C. Tree-Search or Graph-Search Methods

Arenales and Morabito [44] proposed a graph-search approach for 2D nonguillotine cutting problems involving orthogonal cuts. Their method represents first-order cutting patterns using an AND/OR graph, which allows different spatial configurations to be explored systematically. As shown in Fig. 4, a binary-tree search is used to evaluate alternative configurations at each decision node, where each cutting operation may produce either two or five sub-rectangles. The search combines backtracking with hill-climbing to explore promising solutions while allowing alternative configurations to be revisited when needed. To reduce unnecessary exploration, symmetry-elimination rules were applied to remove redundant cutting patterns.

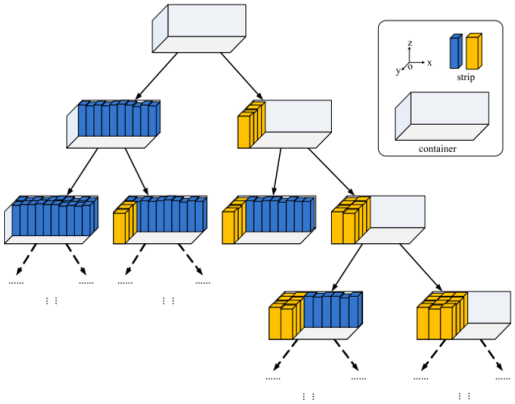


Fig. 4. Binary-tree-based search for efficient container loading plans [45].

D. Conventional Heuristics

Bischoff and Ratcliff [39] identified significant limitations in the existing literature on container loading, arguing that conventional constructive and improvement methods are insufficient for many real-world applications. They emphasized that fundamentally new approaches are required to address practical challenges effectively. Furthermore, they highlighted the necessity of rigorous testing and evaluation to determine the strengths and weaknesses of these new methods, ensuring meaningful comparisons. To support this, they recommended that testing procedures should be standardized as much as possible.

III. Harris Hawks Optimization Algorithm

Heidari et al. [46] introduced the Harris Hawks Optimization (HHO) algorithm, inspired by the cooperative hunting strategies of Harris' hawks in nature. HHO is an innovative, swarm-based metaheuristic, renowned for both its computational efficiency and effectiveness. It utilizes a set of deceptively simple yet highly efficient exploration and exploitation operators, all governed by a dynamically adaptive structure, which makes it particularly well-suited for solving continuous optimization problems.

The HHO algorithm is structured around two core phases, reflecting the hunting tactics of Harris' hawks: exploration and exploitation. The exploration phase emulates the birds' initial search patterns, where candidate solutions are generated through stochastic processes to ensure comprehensive coverage of the search space. This transitions into the exploitation phase, which reflects the hawks' refined pursuit strategies, concentrating computational resources on regions exhibiting promising fitness values while progressively refining solutions. Fig. 5 provides a detailed schematic representation of these interconnected optimization phases, illustrating their respective operational mechanisms and the dynamic transition between broad exploration and focused exploitation in the HHO algorithm [46].

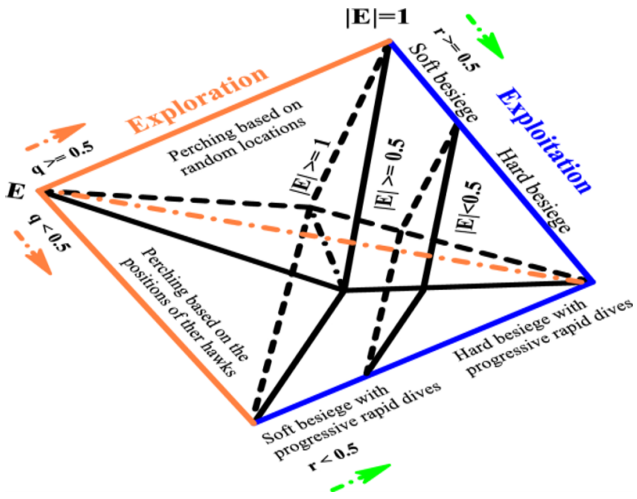


Fig. 5. Phases of HHO [46].

A. Exploration Phase

During this phase, Harris' hawks demonstrate random perching and patient observation to locate prey. Two perching strategies are selected based on a random value q, considering the positions of both family members and the prey. In the exploration phase, q is equally likely for either strategy. When $q < 0.5$, hawks adjust their perch according to the relative positions of their specifics and the rabbit. Alternatively, Harris's hawks perch on randomly selected locations within their home range. The exploration phase's updated position is represented in Eq. (2):

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1|X(t) - 2r_2X_i(t)|, & q \geq 0.5 \\ X_{\text{best}}(t) - X_{\text{mean}}(t) - r_3(L + r_4(U - L)), & q < 0.5 \end{cases} \quad (2)$$

The variables and terms used in the model are defined as follows:

- q: A random number $\in [0, 1]$.
- $X(t)$: Position of i-th hawk at the t-th iteration.
- $X_{\text{best}}(t)$: The position of the rabbit at the t-th iteration.
- r_1, r_2, r_3, r_4: Random numbers $\in [0, 1]$, updated in every iteration.
- L, U: Lower and upper bounds of variables.
- $X_{\text{rand}}(t)$: Selected random hawk from the population.

Eq. (3) represents the mean position of the current hawk population, with N indicating the total number of hawks:

$$X_{\text{mean}}(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{3}$$

B. Transition from Exploration to Exploitation Phase

The Harris Hawks Optimization algorithm incorporates a transfer from an exploration phase to an exploitation phase, depending on the prey's escaping energy. In this algorithm, the prey's energy is represented as gradually decreasing during its escape behavior:

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{4}$$

As in Eq. (4), the energy of the escaping prey, defined as E and initialized as E_0, controls the selection of either exploration or exploitation in the HHO algorithm. In case $|E| \geq 1$, the algorithm is in the exploration phase, while $|E| < 1$ indicates exploitation. E_0 is applied using Eq. (5):

$$E_0 = 2 \times \text{rand} - 1 \tag{5}$$

C. Exploitation Phase

In the exploitation stage, four unique chasing and attacking tactics are applied according to the prey's escaping energy and the hawks' pursuit patterns. The parameter r is used to select a chasing strategy depending on whether the prey successfully escapes ($r < 0.5$) or not ($r \geq 0.5$) before an attack.

D. Soft Besiege

When the probability of escape ($r \geq 0.5$) and the escaping energy ($|E| \geq 0.5$), the prey still possesses sufficient energy and attempts to escape. In response, the Harris' hawks softly surround the prey to deplete its remaining energy before launching an attack. The behavior of the Harris' hawks in this phase is modeled as follows:

$$X(t+1) = \Delta X(t) - E|JX_{\text{best}}(t) - X(t)| \tag{6}$$

$$\Delta X(t) = X_{\text{best}}(t) - X(t) \tag{7}$$

$$J = 2(1 - r_5) \tag{8}$$

Here, $\Delta X(t)$ explains the difference between the current prey's position and its position. J refers to a variable defined as random jump strength.

E. Hard Besiege

When the probability of escape ($r \geq 0.5$) and the escaping energy ($|E| < 0.5$), the prey's energy is low and the Harris' hawks readily encircle it before launching an attack. The positions of the hawks and prey are recalculated using the equation below:

$$X(t+1) = X_{\text{best}}(t) - E|\Delta X(t)| \tag{9}$$

Fig. 6 illustrates this process, modeled through the interaction between a single hawk and the prey.

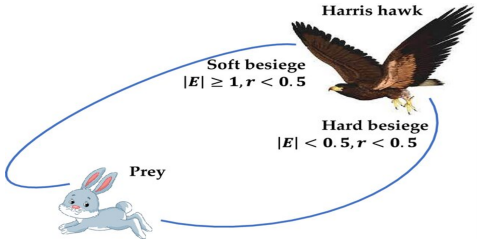


Fig. 6. Vector representation of the hard besiege mechanism [47].

F. Soft Besiege with Progressive Rapid Dives

When the prey's escape energy is high ($|E| \geq 0.5$) and ($r < 0.5$), the hawks carry out a gentle siege, conducting successive quick dives to incrementally refine the prey's location and movement direction. This behavior is modeled using the following equations:

$$Y = X_{\text{best}}(t) - E|JX_{\text{best}}(t) - X(t)| \tag{10}$$

$$Z = Y + S \times LF(D) \tag{11}$$

Where, D is the problem dimension, S is a $1 \times D$ random vector, and LF is the Lévy flight function, calculated via Eq. (12):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \tag{12}$$

where, u and v are random values uniformly distributed in $(0, 1)$, and β is a constant with a default value of 1.5.

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \tag{13}$$

In this context, S represents a random vector, and the superior position between Y and Z is chosen as the subsequent location.

G. Hard Besiege with Progressive Rapid Dives

If ($|E| < 0.5$) and ($r < 0.5$), implying that the prey lacks sufficient energy to evade capture, the hawks execute a hard encirclement by reducing the distance between their mean position and the prey. This action is represented by the following equations:

$$Y = X_{\text{best}}(t) - E|JX_{\text{best}}(t) - X_{\text{mean}}(t)| \tag{14}$$

$$Z = Y + S \times LF(D) \tag{15}$$

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \tag{16}$$

Here, only the more optimal position between Y and Z is adopted as the next point. Algorithm 1 presents the standard HHO pseudocode, summarizing its main optimization steps and position-updating process. Fig. 7 depicts the overall HHO optimization framework, highlighting the exploration and exploitation phases and the associated hawk position-update mechanisms.

Algorithm 1 Pseudocode of HHO algorithm

Require: The population size N and maximum number of iterations T
Ensure: The location of rabbit and its fitness value
Initialize the random population $X_i = (i = 1, 2, \ldots, N)$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set X_{best} as the location of rabbit (best location)
  for (each hawk (X_i)) **do**
    Update the initial energy E_0 and jump strength J
    $E_0 = 2\text{rand}() - 1, J = 2(1 - \text{rand}())$
    Update the E using Eq. (4)
    if ($|E| \geq 1$) **then** ▷ Exploration phase
      Update the location vector using Eq. (2)
    end if
    if ($|E| < 1$) **then** ▷ Exploitation phase
      if ($r \geq 0.5$ **and** $|E| \geq 0.5$) **then** ▷ Soft besiege
        Update the location vector using Eq. (6)
      else if ($r \geq 0.5$ **and** $|E| < 0.5$) **then** ▷ Hard besiege
        Update the location vector using Eq. (9)
      else if ($r < 0.5$ **and** $|E| \geq 0.5$) **then** ▷ Soft besiege with progressive rapid dives
        Update the location vector using Eq. (13)
      else if ($r < 0.5$ **and** $|E| < 0.5$) **then** ▷ Hard besiege with progressive rapid dives
        Update the location vector using Eq. (16)
      end if
    end if
  end for
end while
return X_{rabbit}

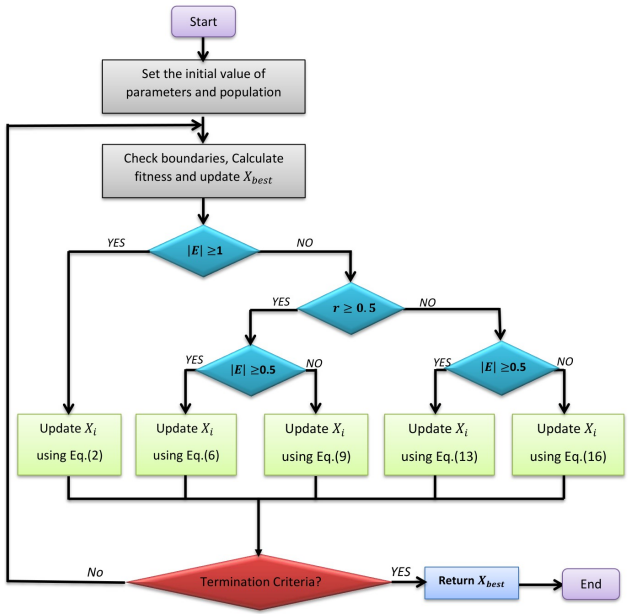


Fig. 7. HHO flowchart.

IV. The Proposed Algorithm (EHHITS)

The complexity of the 3D Container Loading Problem necessitates a robust optimization approach capable of balancing global exploration with precise local refinement. This section details the proposed hybrid framework, which strategically integrates the HHO with an intensive Tabu Search (TS) mechanism, specifically tailored to maximize space utilization. The optimization process initiates with a pre-processing stage where the physical constraints of the container (L, W, H) and box dimensions are established. To ensure computational efficiency, the algorithm pre-calculates all unique spatial orientations for each box type, effectively pruning redundant search paths. To push the boundaries of packing density, the global search is augmented by an intensive Tabu Search, which acts as a high-precision intensification engine. Triggered periodically (every five iterations), this stage focuses exclusively on refining the current global best by navigating its immediate neighborhood through two primary structured moves:

1) Sequence swapping: Modifies the relative priority of items by exchanging their positions within the current packing sequence. This neighborhood operation enables the algorithm to explore alternative item-order configurations, which may lead to more efficient spatial arrangements, improved container utilization, and reduced unused space.

2) Gaussian mutation: Introduces subtle perturbations to fine-tune box orientations and placement coordinates, ensuring no potential spatial optimization is overlooked. To maintain search efficiency, the TS employs a Tabu List, which serves as a short-term memory to prevent the algorithm from cycling back to recently explored, unproductive configurations. However, this restriction is governed by an aspiration criterion: if a "tabu" move yields a utilization higher than the current fitness value.

A. The Robust Multi-Decoder and Extreme Point Logic

The proposed framework centers on a Robust Multi-Decoder—a specialized mechanism that bridges continuous optimization with discrete spatial constraints. Unlike conventional decoders that rely on a single heuristic, this approach evaluates each candidate solution through two competing packing paradigms to identify the most efficient configuration:

1) Vertical layering strategy (column-based packing): Focuses on vertical stratification, prioritizing the stacking of items to maximize layered stability and container height utilization.

2) Horizontal floor-filling strategy (wall-based packing): Focuses on floor-area coverage using a foundation-first approach to promote stable weight distribution.

The fitness value is the maximum utilization achieved by the evaluated packing strategies. An Extreme Point (EP) strategy manages the 3D packing space by dynamically updating feasible insertion points after each box placement, adding new candidate points and removing blocked or infeasible ones. The detailed execution flow and operational steps of the EHHITS algorithm are presented in Algorithm 2, while the pseudocode for the Intensive Tabu Search used during the intensification phase is outlined in Algorithm 3.

Algorithm 2 Pseudocode of the proposed EHHITS algorithm

Require: The population size N and maximum number of iterations T
Ensure: The location of rabbit and its fitness value (Max utilization)
Initialize $X_i = (i = 1, 2, \ldots, N)$
Calculate the fitness values of hawks using Robust Multi Decoder:
fitness values $= \max($core packer Z order, core packer X order$)$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set X_{best} as the location of rabbit (best location)
  for (each hawk (X_i)) **do**
    Update the initial energy E_0 and jump strength J:
    $E_0 = 2\text{rand}() - 1, \quad J = 2(1 - \text{rand}())$
    Update E using Eq. (4)
    if ($|E| \geq 1$) **then** ▷ Exploration phase
      Update the location vector using Eq. (2)
    end if
    if ($|E| < 1$) **then** ▷ Exploitation phase
      Update location vector using Soft/Hard besiege or Levy flight (Eq. 6, 9, 13, 16)
    end if
  end for
  Apply boundary control and Evaluate new fitness value using Robust Multi Decoder
  if ($t \bmod 5 = 0$) **then** ▷ Intensification phase
    Apply Intensive Tabu Search to the location vector X_{best} and fitness value
    Update X_{best} and fitness value
  end if
end while
return X_{best} (Rabbit Location)

TABLE II. PARAMETER SETTING

Parameter	Value	Description
β (Beta)	1.5	Control factor for the search mechanism
Rabbit Energy	[0,2]	The transition between exploration and exploitation
Iterations	300	The maximum number of cycles the algorithm runs for each problem instance
Population Size	30	The number of individual solutions, or "hawks," used in the search process

Algorithm 3 Pseudo-code of Intensive Tabu Search

Require: Best location X_{best}, Current fitness, local search iterations (T_{iter})
Ensure: Refined X_{best} and its fitness value (Rabbit Energy)
Set current$_x = X_{\text{best}}$
for $j = 1$ **to** T_{iter} **do**
  Generate Neighbor hawk by applying Swap and Mutation on current$_x$
  Calculate fitness neighbor using Robust Multi Decoder
  if (fitness$_{\text{neighbor}} >$ Rabbit Energy) **then**
    Update $X_{\text{best}} = X_{\text{Neighbor}}$ and Rabbit Energy $=$ fitness$_{\text{neighbor}}$
  end if
end for
return X_{best}, fitness value (Rabbit Energy)

V. COMPUTATIONAL RESULTS

To evaluate the performance of the proposed method, this study employs the standard 15 LN instances (LN1 to LN15) introduced by Loh and Nee [48] as well as the 700 instances from Bischoff and Ratcliff [49]. These test instances vary in the number of box types and 3D bin packing dimensions, spanning from weakly to strongly heterogeneous problem characteristics. The proposed EHHITS algorithm was implemented in MATLAB® 2018 and executed on a laptop computer equipped with an Intel® Core i5 CPU (4 GHz) and 16 GB of RAM. The performance of the EHHITS algorithm for the test cases was governed by the following critical parameters as detailed in Table II:

A. Results for Test Cases LN01–LN15

For the performance evaluation, EHHITS is applied to a 3D bin packing data set. The Loh and Nee (LN) data set [48] is known for its heterogeneous nature, meaning it contains a wide variety of different box sizes and types. This makes the packing problem particularly difficult, as algorithms must be flexible enough to handle many different shapes and quantities of items, rather than just a few uniform sizes. The LN data set for the 3D container loading problem consists of 15 problems. This data set is notable for its heterogeneous nature, meaning each problem involves a variety of different box sizes and types, which makes it a challenging benchmark for packing algorithms.

The instances in the LN data set typically involve:

- A single container of a defined size.

TABLE III. VOLUME UTILIZATION (%) FOR TEST CLASSES LOH AND NEE (LN)

Problem	[48]	[50]	[51]	[49]	[40]	[2]	[41]	[3]	[52]	[53]	[54]	EHHITS (This work)
LN1	78.1	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5
LN2	76.8	80.7	89.7	90.0	93.9	89.8	90.8	80.4	96.4	93.2	94.6	96.6
LN3	69.5	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4
LN4	59.2	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0
LN5	85.8	77.2	77.2	77.2	77.2	77.2	77.2	76.7	77.2	77.2	77.2	77.2
LN6	88.6	88.7	89.5	83.1	91.6	92.4	87.9	84.8	93.5	93.0	92.6	95.4
LN7	78.2	81.8	83.9	78.7	84.7	84.7	84.7	77.0	84.7	84.7	84.7	84.7
LN8	67.6	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4
LN9	84.2	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9
LN10	70.1	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3
LN11	63.8	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2
LN12	76.3	78.5	76.5	78.5	78.5	78.5	78.5	69.5	78.5	78.5	78.5	78.5
LN13	77.0	84.1	82.3	78.1	85.1	85.6	85.6	73.3	84.9	85.6	85.6	85.6
LN14	69.1	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8
LN15	65.6	59.6	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5
Average	74.2	69.0	69.5	68.6	70.3	69.9	69.9	67.05	70.6	70.4	70.5	70.8
Number of optimal problems	11	11	10	11	12	13	13	11	12	13	13	13

TABLE IV. VOLUME UTILIZATION (%) FOR TEST CLASSES BR1–BR7

Problem	Maximum	Minimum	Average	SD
BR1(3)	95.58	86.42	91.24	2.14
BR2(5)	95.36	86.57	91.86	1.82
BR3(8)	94.10	87.10	92.14	1.54
BR4(10)	93.42	86.50	92.72	1.43
BR5(12)	93.44	87.30	91.57	1.56
BR6(15)	93.23	88.46	91.42	1.39
BR7(20)	92.40	86.23	91.12	1.42

TABLE V. AVERAGE VOLUME UTILIZATION (%) FOR TEST CLASSES BR1–BR7

Approach	Volume utilization (%)
[40]	88.51
[41]	88.75
[51]	84.94
[52]	91.81
[54]	84.56
[49]	83.45
[55]	88.30
[56]	91.26
[57]	92.86
[58]	89.73
[59]	93.82
[60]	84.19
EHHITS (this work)	91.72

- Multiple types of rectangular boxes, with each type having different dimensions (length, width, height) and a specified quantity to be packed.
- The objective is to optimize the placement of boxes in the container for highest volume efficiency.

The constraints of the 3D bin packing are as follows:

- The items consist of rectangular boxes characterized by their known width, depth, and height.
- The boxes can be reoriented if permitted by problem constraints.
- Each box is allowed to be stacked over another.

To preserve structural orientation in these particular situations, rotations along the horizontal axes were restricted. The dataset used is made up of 100-250 instances that are divided into 6-10 different kinds. The EHHITS method was executed for a fixed duration of 10 minutes for each instance. To ensure a reliable evaluation and account for the stochastic nature of the metaheuristic, each instance was independently solved in 30 runs. This repeated evaluation helps reduce the effect of random initialization and provides a more robust assessment of the algorithm's performance.

Table III presents the experimental results for the 15 Loh/Nee test cases together with those found in the literature. The obtained volume utilization is provided for every method except L/N. According to Bischoff and Ratcliff [49], the packing density expressed in L/N is an overestimation of the volume utilization. Lastly, for each strategy, the average volume utilization and the number of ideal solutions for every box packed are provided. Our algorithm produced the highest average volume utilization for the 15 test scenarios out of all the methods. Fig. 8 presents samples of the packing results obtained by EHHITS on the LN benchmark dataset (LN2, LN6, LN12 and LN13).

The experimental results demonstrate that the EHHITS algorithm is a highly effective metaheuristic for solving the 3D packing problem. The algorithm successfully packed all available items in 13 out of 15 benchmark instances. For the remaining two instances, EHHITS achieved volume utilization

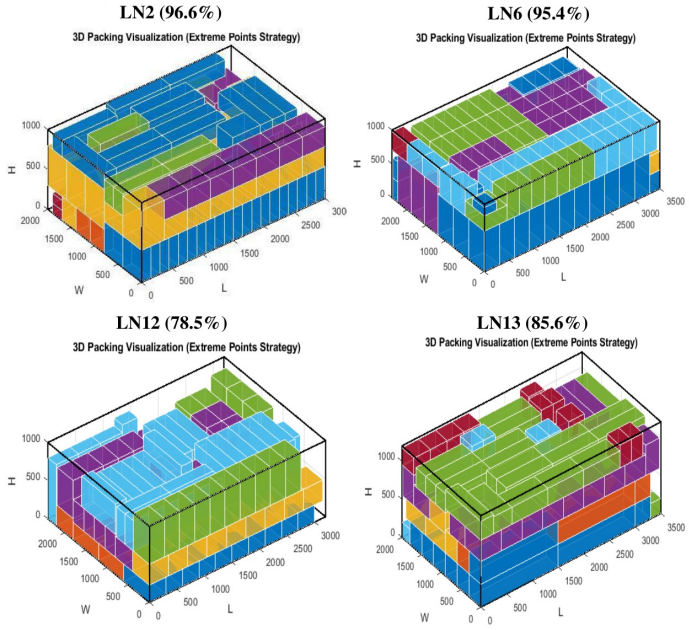


Fig. 8. LN dataset packing results.

rates of 96.6% and 95.4%, respectively, yielding an average utilization rate of 70.8% across the evaluated benchmark instances.

B. Results for Test Classes BR1–BR7

The 700 Bischoff/Ratcliff test cases are organized into seven test classes (BR1-BR7), with each class consisting of 100 test cases. For all test scenarios, the container dimensions are L = 587, W = 233, and H = 220. The number of box kinds for BR1 through BR7 is 3, 5, 8, 10, 12, 15, and 20. Table IV summarizes the experimental results for the 700 Bischoff/Ratcliff test cases. For every test class, the minimum, average, and maximum volume utilizations of 100 test instances are displayed together with the instances' standard deviations. The EHHITS algorithm is executed for ten minutes and 30 times with various random numbers to solve each problem. The total average volume use for the 700 test instances is shown in Table V and contrasted with the utilizations attained using alternative techniques. Other methods' volume utilizations are derived from the literature and updated with the latest findings. The results of EHHITS indicate that for all test classes, the average volume utilization is 91.72%.

Several factors intrinsic to the EHHITS design contribute to this robust performance. Firstly, the hybridization strategy successfully balances global exploration and local intensification. The multi-strategy initialization generates a diverse and high-quality initial population, allowing the HHO component to effectively search for optimal regions in the solution space. Subsequently, the Intensive Tabu Search acts as a powerful intensifier, refining these solutions by navigating complex local optima and escaping from suboptimal regions through its diverse neighborhood operators. This synergistic interaction is likely the primary driver behind the algorithm's consistency and reliability, as reflected in its high success rate across most test instances.

VI. Conclusion

This research has introduced and empirically validated a novel hybrid metaheuristic for addressing the 3D container packing problem, combining Enhanced Harris Hawks Optimization with Tabu Search (EHHITS). The proposed algorithm addresses fundamental challenges in complex combinatorial optimization by integrating global exploration capabilities with intensive local search techniques. The EHHITS framework integrates an Intensive Tabu Search with HHO to strategically balance local precision and global exploration to increase packing density. Comprehensive evaluation on the standard LN benchmark instances demonstrates the strong and competitive performance of EHHITS. The proposed algorithm achieved optimal solutions in 13 of the 15 test cases and attained an average volume utilization of 70.8%, with notable improvements observed on the LN2 and LN6 instances. Furthermore, evaluation on the complete set of 700 Bischoff/Ratcliff test cases demonstrates the robustness and scalability of EHHITS across packing instances with varying problem dimensions. For all test classes ranging from BR1 to BR7, the average volume utilization consistently remains above 91%. These findings, supported by the detailed metrics of minimum, maximum, and standard deviation values, suggest that the integration of Tabu Search into the Enhanced Harris Hawks framework significantly strengthens global search capabilities. Ultimately, the EHHITS algorithm proves to be a reliable and high-performing strategic framework for achieving superior global optimization in diverse packing environments. The experimental outcomes confirm that the strategic hybridization of population-based and trajectory-based methods offers significant advantages for tackling NP-hard optimization problems. Future research directions include developing dynamic parameter adaptation

mechanisms, incorporating additional problem-specific constraints, and extending the approach to other combinatorial optimization domains.

ACKNOWLEDGMENT

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-23-DR-275). Therefore, the authors thank the University of Jeddah for its technical and financial support.

REFERENCES

[1] G.Scheithauer,"Algorithms for the Container Loading Problem," in *Operations Research Proceedings 1991*, W.Gaul *et al.*, Eds. Berlin, Heidelberg: Springer, 1992, pp. 445-452, doi:10.1007/978-3-642-46773-8-112.

[2] A.Bortfeldt and H.Gehring, "A hybrid genetic algorithm for the container loading problem," *European Journal of Operational Research*, vol. 131, no. 1, pp. 143–161, 2001, doi: 10.1016/S0377-2217(00)00055-2.

[3] T.Dereli and G.S.Das, "A hybrid 'bee(s) algorithm' for solving container loading problems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2854–2862, 2011, doi: 10.1016/j.asoc.2010.11.017.

[4] S.K.ElSayed and E.E.Elattar, "Hybrid Harris hawks optimization with sequential quadratic programming for optimal coordination of directional overcurrent relays incorporating distributed generation," *Alexandria Engineering Journal*, vol. 60, no. 2, pp. 2421–2433, 2021, doi: 10.1016/j.aej.2020.12.028.

[5] Z. Yu, X. Shi, J. Zhou, X. Chen, and X. Qiu, "Effective Assessment of Blast-Induced Ground Vibration Using an Optimized Random Forest Model Based on a Harris Hawks Optimization Algorithm," *Applied Sciences*, vol. 10, no. 4, p. 1403, 2020, doi: 10.3390/app10041403.

[6] M.A.Al-Betar, M.A.Awadallah, A.A.Heidari, H.Chen *et al.*, "Survival exploration strategies for Harris Hawks Optimizer," *Expert Systems with Applications*, vol. 168, p. 114243, 2021, doi: 10.1016/j.eswa.2020.114243.

[7] M.H.Qais, H.M.Hasanien, and S.Alghuwainem, "Parameters extraction of three-diode photovoltaic model using computation and Harris Hawks optimization," *Energy*, vol. 195, p. 117040, 2020, doi: 10.1016/j.energy.2020.117040.

[8] S.Jiao, C.Wang, R.Gao, Y.Li, and Q.Zhang, "Harris Hawks Optimization with Multi-Strategy Search and Application," *Symmetry*, vol. 13, no. 12, p. 2364, 2021, doi: 10.3390/sym13122364.

[9] E. Rodríguez-Esparza *et al.*, "An efficient Harris hawks-inspired image segmentation method," *Expert Systems with Applications*, vol. 155, p. 113428, 2020, doi: 10.1016/j.eswa.2020.113428.

[10] W.Fu, K.Shao, J.Tan, and K.Wang, "Fault Diagnosis for Rolling Bearings Based on Composite Multiscale Fine-Sorted Dispersion Entropy and SVM With Hybrid Mutation SCA-HHO Algorithm Optimization," *IEEE Access*, vol. 8, pp. 13086–13104, 2020, doi: 10.1109/ACCESS.2020.2966582.

[11] H.Moayedi, D.Bui, and P.ThiNgo, "Neural Computing Improvement Using Four Metaheuristic Optimizers in Bearing Capacity Analysis of Footings Settled on Two-Layer Soils," *Applied Sciences*, vol. 9, no. 23, p. 5264, 2019, doi: 10.3390/app9235264.

[12] M.Abdel-Basset, W.Ding, and D.El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 593–637, 2021, doi: 10.1007/s10462-020-09860-3.

[13] A.Abbasi, B.Firouzi, and P.Sendur, "On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks," *Engineering with Computers*, vol. 37, no. 2, pp. 1409–1428, 2021, doi: 10.1007/s00366-019-00892-0.

[14] K.Shao, W.Fu, J.Tan, and K.Wang, "Coordinated approach fusing time-shift multiscale dispersion entropy and vibrational Harris hawks optimization-based SVM for fault diagnosis of rolling bearing," *Measurement*, vol. 173, p. 108580, 2021, doi: 10.1016/j.measurement.2020.108580.

[15] N.Kaur, L.Kaur, and S.S.Cheema, "An enhanced version of Harris Hawks Optimization by dimension learning-based hunting for Breast Cancer Detection," *Scientific Reports*, vol. 11, no. 1, p. 21933, 2021, doi: 10.1038/s41598-021-01018-7.

[16] M.Naeijian, A.Rahimnejad, S.M.Ebrahimi, N.Pourmousa, and S.A.Gadsden, "Parameter estimation of PV solar cells and modules using Whippy Harris Hawks Optimization Algorithm," *Energy Reports*, vol. 7, pp. 4047–4063, 2021, doi: 10.1016/j.egyr.2021.06.085.

[17] L.K.Foong and H.Moayedi, "Slope stability evaluation using neural network optimized by equilibrium optimization and vortex search algorithm," *Engineering with Computers*, vol. 38, no. S2, pp. 1269–1283, 2022, doi: 10.1007/s00366-021-01282-1.

[18] İ. Çetınbaş, B. Tamyürek, and M. Demırtaş, "The Hybrid Harris Hawks Optimizer-Arithmetic Optimization Algorithm: A New Hybrid Algorithm for Sizing Optimization and Design of Microgrids," *IEEE Access*, vol. 10, pp. 19254–19283, 2022, doi: 10.1109/ACCESS.2022.3151119.

[19] A. G. Hussien and M. Amin, "A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 2, pp. 309–336, 2022, doi: 10.1007/s13042-021-01326-4.

[20] H. Zhang *et al.*, "A generalized artificial intelligence model for estimating the friction angle of clays in evaluating slope stability using a deep neural network and Harris Hawks optimization algorithm," *Engineering with Computers*, vol. 38, no. S5, pp. 3901–3914, 2022, doi: 10.1007/s00366-020-01272-9.

[21] R. M. Rabie *et al.*, "An Improved Harris Hawks Optimizer (IHHO) with Adaptive Hill Climbing and Elite Mutation for Global Optimization," *Results in Control and Optimization*, 2026, doi: 10.1016/j.rico.2026.100815.

[22] R.M.Rabie, H.Zaher, N.R.Saied, and H.Sayed, "A Modified Harris Hawks Algorithm to Solving Optimization Problems," *International Journal of Industrial Engineering & Production Research*, vol. 35, no. 1, 2024, doi: 10.22068/ijiepr.35.1.1853.

[23] A. Dabba, A. Tari, and S. Meftali, "A new multi-objective binary Harris Hawks optimization for gene selection in microarray data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3157–3176, 2023, doi: 10.1007/s12652-021-03441-0.

[24] H. M. Balaha, A. E.-S. Hassan, E. M. El-Gendy, H. ZainEldin, and M. M. Saafan, "An aseptic approach towards skin lesion localization and grading using deep learning and harris hawks optimization," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 19787–19815, 2023, doi: 10.1007/s11042-023-16201-3.

[25] D. Dhawale, V. K. Kamboj, and P. Anand, "An improved Chaotic Harris Hawks Optimizer for solving numerical and engineering optimization problems," *Engineering with Computers*, vol. 39, no. 2, pp. 1183–1228, 2023, doi: 10.1007/s00366-021-01487-4.

[26] S. Almotairi, E. Badr, M. Abdul Salam, and A. Dawood, "Three Chaotic Strategies for Enhancing the Self-Adaptive Harris Hawk Optimization Algorithm for Global Optimization," *Mathematics*, vol. 11, no. 19, p. 4181, 2023, doi: 10.3390/math11194181.

[27] C.Li, Q.Si, J.Zhao, and P.Qin, "A robot path planning method using improved Harris Hawks optimization algorithm," *Measurement and Control*, vol. 57, no. 4, pp. 469–482, 2024, doi: 10.1177/00202940231204424.

[28] Q.V. Vu, V.T. Pham, D.N. Le, Z. Kong, G. Papazafeiropoulos *et al.*, "Hybrid machine learning with HHO method for estimating ultimate shear strength of both rectangular and circular RC columns," *Steel and Composite Structures*, vol. 52, no. 2, pp. 145–163, 2024, doi: 10.12989/SCS.2024.52.2.145.

[29] F. Wang, "Enhanced Task Scheduling Algorithm Using Harris Hawks Optimization Algorithm for Cloud Computing," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, 2025, doi: 10.14569/IJACSA.2025.0160189.

[30] M. reza Behnamfar, H. Barati, M. Abasi, M. Alaleh, and G. A. Torabi, "Multi-Objective Optimization for Day-Ahead HT-WP-PV-PSH with LS-EVs Systems Self-Scheduling Unit Commitment Using HHO-PSO Algorithm," *Journal of Operation and Automation in Power Engineering*, Online First, 2025, doi: 10.22098/joape.2024.12419.1934.

[31] R. Ghadami and J. Rahebi, "Alzheimer's Prediction Methods with Harris Hawks Optimization (HHO) and Deep Learning-Based Approach Using an MLP-LSTM Hybrid Network," *Diagnostics*, vol. 15, no. 3, p. 377, 2025, doi: 10.3390/diagnostics15030377.

[32] S. Al-Adwan *et al.*, "A multi-strategy framework for enhancing Harris hawks optimization for global optimization problems," *Scientific Reports*, 2026.

[33] M. Kumar, N. Mohd, G. Shivam, A. Goyal, D. Parashar *et al.*, "Hybrid Aquila optimizer–Harris Hawks optimization for CNN hyperparameter tuning in brain tumor classification," *Scientific Reports*, 2026.

[34] A. Aliano Filho, H. De Oliveira Florentino, M. V. Pato, S. C. Poltroniere, and J. F. Da Silva Costa, "Exact and heuristic methods to solve a bi-objective problem of sustainable cultivation," *Annals of Operations Research*, vol. 314, no. 2, pp. 347-376, 2022, doi:10.1007/s10479-019-03468-9.

[35] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003, doi: 10.1145/937503.937505.

[36] J. F. Gonçalves and M.G.C.Resende, "A parallel multi-population biased random-key genetic algorithm for a container loading problem," *Computers & Operations Research*, vol. 39, no. 2, pp. 179–190, 2012, doi: 10.1016/j.cor.2011.03.009.

[37] R.Zeineldin and A.Morsy, "A Modified Artificial Bee Colony for Solving the Container Loading Problem," *International Journal of Computer Applications*, vol. 114, no.3, pp. 19-24, 2015, doi:10.5120/19958-1786.

[38] T. Fanslau and A. Bortfeldt, "A Tree Search Algorithm for Solving the Container Loading Problem," *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 222–235, 2010, doi: 10.1287/ijoc.1090.0338.

[39] E. E. Bischoff and M. S. W. Ratcliff, "Issues in the development of approaches to container loading," Omega, vol. 23, no. 4, pp. 377–390, Aug. 1995, doi: 10.1016/0305-0483(95)00015-G.

[40] J.Terno, G.Scheithauer, U.Sommerweiß, and J.Riehme, "An efficient approach for the multi-pallet loading problem," *European Journal of Operational Research*, vol. 123, no. 2, pp. 372–381, 2000, doi: 10.1016/S0377-2217(99)00263-5.

[41] M. Eley, "Solving container loading problems by block arrangement," *European Journal of Operational Research*, vol. 141, no. 2, pp. 393–409, 2002, doi: 10.1016/S0377-2217(02)00133-9.

[42] R. Morabito, "An AND/OR-graph approach to the container loading problem," Int. Trans. Oper. Res., vol. 1, no. 1, pp. 59–73, Jan. 1994, doi: 10.1016/0969-6016(94)90046-9.

[43] D. T. Akl, M. M. Saafan, A. Y. Haikal, and E. M. El-Gendy, "IHHO: an improved Harris Hawks optimization algorithm for solving engineering problems," *Neural Computing and Applications*, vol. 36, no. 20, pp. 12185-12298, 2024, doi:10.1007/s00521-024-09603-3.

[44] M. Arenales and R. Morabito, "An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems," Eur. J. Oper. Res., vol. 84, no. 3, pp. 599–617, Aug. 1995, doi: 10.1016/0377-2217(95)00026-M.

[45] S. Liu, W. Tan, Z. Xu, and X. Liu, "A tree search algorithm for the container loading problem," Comput. Ind. Eng., vol. 75, pp. 20–30, Sep. 2014, doi: 10.1016/j.cie.2014.05.024.

[46] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja *et al.*, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.

[47] H. M. Alabool, D. Alarabiat, L. Abualigah, and A. A. Heidari, "Harris hawks optimization: a comprehensive review of recent variants and applications," Neural Comput. Appl., vol. 33, no. 15, pp. 8939–8980, Aug. 2021, doi: 10.1007/s00521-021-05720-5.

[48] T. Loh and A. Nee, "A packing algorithm for hexahedral boxes," in *Proc. Conf. Ind. Automation*, Singapore, 1992, pp. 115-126.

[49] E.E.Bischoff and M.S.W.Ratcliff, "Loading Multiple Pallets," *Journal of the Operational Research Society*, vol. 46, no. 11, pp. 1322-1336, 1995, doi:10.1057/jors.1995.181.

[50] B. K. A. Ngoi, M. L. Tay, and E. S. Chua, "Applying spatial representation techniques to the container packing problem," *International Journal of Production Research*, vol. 32, no. 1, pp. 111–123, 1994, doi:10.1080/00207549408956919.

[51] E. E. Bischoff, F. Janetz, and M. S. W. Ratcliff, "Loading pallets with non-identical items," *European Journal of Operational Research*, vol. 84, no. 3, pp. 681–692, 1995, doi:10.1016/0377-2217(95)00031-K.

[52] A. Lim, H. Ma, J. Xu, and X. Zhang, "An iterated construction approach with dynamic prioritization for solving the container loading problems," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4292-4305, 2012, doi:10.1016/j.eswa.2011.09.103.

[53] T. A. M. Toffolo, E. Esprit, T. Wauters, and G. Vanden Berghe, "A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem," *European Journal of Operational Research*, vol. 257, no. 2, pp. 526–538, 2017, doi:10.1016/j.ejor.2016.07.033.

[54] Ö. Şafak and G. Erdoğan, "A Large Neighbourhood Search Algorithm for Solving Container Loading Problems," *Computers & Operations Research*, vol. 154, p. 106199, 2023, doi:10.1016/j.cor.2023.106199.

[55] H. Gehring and A. Bortfeldt, "A Genetic Algorithm for Solving the Container Loading Problem," *International Transactions in Operational Research*, vol. 4, no. 5–6, pp. 401–418, 1997, doi:10.1111/j.1475-3995.1997.tb00095.x.

[56] A. Bortfeldt and H. Gehring, "Applying Tabu Search to Container Loading Problems," in *Operations Research Proceedings 1997*, P. Kischka *et al.*, Eds. Berlin, Heidelberg: Springer, 1998, pp. 533–538, doi:10.1007/978-3-642-58891-4-84.

[57] D. Mack, A. Bortfeldt, and H. Gehring, "A parallel hybrid local search algorithm for the container loading problem," Int. Trans. Oper. Res., vol. 11, no. 5, pp. 511–533, Sep. 2004, doi: 10.1111/j.1475-3995.2004.00474.x.

[58] A. Moura and J. F. Oliveira, "A GRASP Approach to the Container-Loading Problem," IEEE Intell. Syst., vol. 20, no. 4, pp. 50–57, Jul. 2005, doi: 10.1109/MIS.2005.57.

[59] F. Parreño, R. Alvarez-Valdes, J. M. Tamarit, and J. F. Oliveira, "A Maximal-Space Algorithm for the Container Loading Problem," Inf. J. Comput., vol. 20, no. 3, pp. 412–422, Aug. 2008, doi: 10.1287/ijoc.1070.0254.

[60] A. P. Davies and E. E. Bischoff, "Weight distribution considerations in container loading," Eur. J. Oper. Res., vol. 114, no. 3, pp. 509–527, May 1999, doi: 10.1016/S0377-2217(98)00139-8.

Energy Personalized Recommender for the PowerEye Home Energy Management System

Reem Hejazi*, Sofianiza Abd Malik, Anees Ara, Layla Alfawzan
College of Computer and Information Science, Prince Sultan University, Riyadh, Saudi Arabia

***Abstract*—The Energy Personalized Recommender (EPR) is a novel backend module designed to optimize energy management within home energy management systems (HEMSs). Existing research explores various energy optimization strategies but is often constrained by synthetic datasets, simulation-based validation, or limited personalization. To address these limitations, EPR is designed for real-world deployment and personalization. By employing machine learning algorithms, the EPR analyzes appliance-level energy consumption in real time and integrates various algorithms to enhance user interaction with energy usage data. These algorithms include the Energy Goal Algorithm, which allows users to set and monitor monthly energy cost goals, and the Peak Time Algorithm, which advises on shifting usage to off-peak times. The Phantom Mode Algorithm identifies low-power or standby appliances, promoting energy conservation, while the Energy Baseline Algorithm predicts hourly energy usage to detect deviations. In evaluating the system, k-means clustering for phantom appliances achieved silhouette scores of 0.478 to 0.954, indicating reliable performance. Baseline prediction models showed RMSE values between 0.003 and 1.526, depending on the appliance. A single-day evaluation involving three participants showed mixed consumption changes: two participants recorded decreases, while the third recorded an increase, resulting in a 19.74% aggregate increase. These preliminary findings provide a foundation for advancing personalized energy management and motivate longer controlled trials to determine efficient and sustained energy savings through EPR systems.**

Keywords*—Home energy management systems; machine learning; energy goal; peak time; phantom mode; energy baseline; k-means clustering; XGBoost forecasting; energy sustainability

I. Introduction

Recommender systems have become essential tools in various domains, offering personalized suggestions that enhance user experience and decision-making. In energy management, these systems are crucial in promoting energy efficiency and sustainability. Home energy management systems (HEMSs) are designed to collect and analyze energy data from household appliances, providing users with valuable insights into their energy usage patterns [1].

Current research in the field of HEMS has explored not only basic energy monitoring but also the integration of recommender systems to provide personalized insights [2], [3]. Several studies have proposed hybrid energy recommender systems that combine multiple optimization strategies to support household energy management [4], [5], including demand-side management in smart grids [6] and energy consumption forecasting using machine learning approaches such as RNN-LSTM [7]. While these approaches demonstrate the benefits of hybridization, they typically focus on a limited subset of objectives. In contrast, our proposed HEMS, PowerEye, introduces an Energy Personalized Recommender (EPR) designed as a modular hybrid framework that simultaneously integrates four complementary optimization algorithms: cost-based goal tracking, peak-time shifting, phantom energy detection, and real-time baseline forecasting. This unified design enables more comprehensive and actionable recommendations, thereby enhancing both energy efficiency and sustainability.

This paper aims to explore the development and impact of EPR within the PowerEye system. The contributions of the proposed work are as follows:

- We conduct a detailed state-of-the-art literature review of energy recommender system technologies.
- We propose a hybrid-approach EPR framework as part of the HEMS, PowerEye. This system demonstrates personalized recommendations that can optimize energy usage in households.
- We deploy PowerEye using 23 commercial smart plugs across three households, collecting 111 days of appliance-level data for model development and evaluation. A separate single-day experiment examines consumption changes following recommendations.
- We develop the EPR algorithms using k-means clustering and XGBoost. These algorithms are used to recommend and predict optimal energy usage.
- We conduct an evaluation of the experiments, including phantom clustering accuracy, baseline forecasting accuracy, energy consumption analysis, and user experience. These results assess the effectiveness of personalized recommendations.

The paper is organized as follows: Section II provides a comprehensive review of the literature on energy recommendation systems; Section III outlines the PowerEye system architecture and related details; Section IV describes the simulation setup and deployment of the proposed system; Sections V and VI present the results and discussion, respectively; and finally, Section VII concludes the paper with future recommendations.

II. Background

The following section provides a brief overview of the background of energy recommender systems and a comparative analysis of related work.

*Corresponding author.

A. Types of Energy Recommender Systems

The primary aim of HEMSs is to enable users to monitor and optimize their energy consumption through personalized recommendations. Various models are employed for energy optimization in recommender systems, including the following:

1) Content-based: Suggests items that are similar to those a user has liked in the past by analyzing item features [8]. It builds a user profile based on features of items previously rated by the user, such as keywords from item descriptions.

2) Collaborative filtering: Makes suggestions based on the behaviors of users with similar preferences [9]. This approach analyzes the rating history of users and calculates the similarity between them. Recommendations are then made by suggesting items that other users with similar tastes have liked in the past.

3) Knowledge-based: Suggests items based on predefined rules and domain knowledge, ensuring they meet specific user requirements and preferences [10]. Unlike systems based on rankings or item features, this approach evaluates items or actions against a set of criteria defined by the user.

4) Case-based: Uses historical examples or past cases to inform recommendations [11]. It retrieves relevant cases from memory to solve new problems by adapting previous workflows to fit current conditions. Once a suitable case is identified, it is modified for the new situation and reused.

5) Context-based: Suggests items by considering contextual factors such as time, location, or other external conditions [12]. By adapting to the user's changing needs and situations, context-based systems aim to provide more relevant and timely recommendations that align with the user's current circumstances.

6) Hybrid: Combines two or more recommendation techniques to improve the accuracy and relevance of recommendations.

B. Literature Review

Jimenez-Bravo et al. [10] proposed a multi-agent system for smart homes that optimizes energy usage by coordinating agents. These agents, built with a Python library, collect real-time consumption data, track electricity prices, and offer personalized energy-saving recommendations. The system uses knowledge-based and utility-based recommendations to provide financial incentives for shifting energy use. Though promising for addressing energy optimization issues in the European Union, the system was only tested in a theoretical case study, not real-world settings.

Eirinaki et al. [4] proposed a framework to help households improve energy efficiency by providing real-time recommendations for appliance use. The framework uses non-intrusive load monitoring (NILM) to disaggregate individual appliance consumption from main meter data and applies association rule mining and sequential pattern mining algorithms to extract usage patterns. These patterns are used to build personalized user profiles and generate recommendations for energy savings. The study found that the framework could reduce energy consumption by 2–17%.

Gonzalez-Briones et al. [11] present a multi-agent system architecture designed to optimize temperature and energy consumption in an office building. By analyzing sensor data and external information, the system establishes temperature patterns that align with both energy efficiency and user comfort. The GAIA methodology [13] guided the design and development of this system, incorporating algorithms for savings, machine learning, case-based reasoning, and artificial neural networks. The findings demonstrate the system's effectiveness, achieving average energy savings of 41% in the experimental group offices compared to the control group.

Zhang et al. [14] introduce an electricity plan recommender system (EPRS) designed to offer personalized recommendations to residential customers. Leveraging neighborhood-based collaborative filtering, the EPRS demonstrates superior performance compared to existing strategies in terms of recommendation accuracy. The J-wEUC similarity metric, a novel approach developed for this study, effectively addresses the challenge of missing features. By providing customers with predicted ratings of various tariff plans, the EPRS facilitates the widespread adoption of flexible pricing tariffs.

Sardianos et al. [5] present REHAB-C, a context-aware recommendation system designed to assist users in transforming their energy habits. REHAB-C utilizes a rule-based system to detect user habits and recommend actions based on the user's context and goals. Additionally, the system incorporates collaborative filtering to leverage feedback from multiple users, enhancing the recommendations. REHAB-C was evaluated in real-world office scenarios and demonstrated a significant reduction in energy consumption, up to 50%.

Zhang et al. [15] present a customized electricity retailing plan recommendation strategy that assists customers in selecting economical retailing plans. The strategy utilizes customer electricity usage patterns to recommend cost-effective plans without requiring full daily load profile records. The methodology involved two-stage DLP clustering, correlative feature selection, decision tree classification, and typical plan rank and recommendation plan rank calculation. The study identified correlative features between load profile features and DLP classes, providing insights for both plan selection and further investigations into the retail market, such as tariff design.

Rodriguez et al. [16] present a novel methodology that utilizes various algorithms to analyze energy consumption patterns in household refrigerating appliances, leading to tailored energy-saving recommendations. Their approach incorporates cycle counting to determine operating cycles and a 7-day centered moving average for smoothing energy data and estimating base consumption. By employing k-means clustering with Dynamic Time Warping, the study categorizes daily energy profiles. The findings demonstrate that the probability density estimation method effectively identifies refrigerators with anomalously high energy consumption, enabling personalized recommendations.

A comparison between these existing energy recommendation systems is shown in Table I.

The literature presents diverse frameworks for energy management, including multi-agent systems, real-time recommendations, and personalized plan suggestions. These studies leverage approaches like collaborative filtering, machine learning, and clustering to optimize energy use and provide

TABLE I. COMPARISON OF EXISTING ENERGY RECOMMENDATION SYSTEMS

	Features	Methods	Limitations
[10]	- Minimize peak consumption - Utilize smart plugs - Use online electricity prices	- Multi-agent - Knowledge-based - Utility-based	- Not real-world settings - Limited to the Spanish electricity market - Supports only one smart plug
[4]	- Create appliance profiles - Switch on/off suggestions - Interactive energy suggestion interface	- HMM model - Combinatorial optimization - k-means clustering	- Ignore inter-appliance associations - Omit weather data - Omit environ. psychology
[11]	- Automatic temperature adjustment - Data-driven HVAC temperature control - Case-based reasoning	- Multi-agent - GAIA - Case-based	- Lacks comfort parameters with energy savings - Insufficient data reporting - Lacks granular insights
[14]	- Recommend cost-effective electricity plans - Offer predicted ratings - Appliance on thresholds	- CF filtering - Neighborhood-based - Jaccard Euclidean	- Slow calculation speed - Limited dataset - Limited exploration of the impact on market policies
[5]	- Goal-based system - Incorporate smart sensors - Incorporate users' preferences and feedback	- CF filtering - Rule-based - Context-aware	- No support for setting energy goals - Limited to a single user - No standardized benchmarks
[15]	- Rank cost-effective electricity - Classify daily load profiles - Link DLP classes to a typical plan rank	- k-means clustering - Decision trees - Dominant plan rank selection	Not mentioned
[16]	- Analyze fridge energy use - Evaluate base and relative energy consumption - Identify anomalous energy	- One-way ANOVA - k-means clustering - Probability density estimation	- Limited dataset - Recommender impact has not been evaluated - Does not consider seasonal effects

savings insights. Notable findings show energy reduction potential from 2% to 50%, though many systems are limited by theoretical models, dataset constraints, or lack of real-world validation. Hence, in the following section, we address some of these gaps and propose an energy-personalized recommender for a home energy management system.

III. METHODS

In this section, we discuss the design and architecture of our proposed HEMS, PowerEye, along with its system components, personalized recommendation strategy, and the recommendation process. Finally, we provide an in-depth explanation of the four algorithms used in our EPR: energy goal, peak time, phantom mode, and energy baseline algorithms.

A. PowerEye: The Proposed Home Energy Management System

Our proposed HEMS, PowerEye, aims to build a user-friendly mobile application that empowers users to effortlessly monitor their energy usage at the appliance level. Users can set energy consumption goals, easily manage devices and rooms, and receive real-time data insights to make informed decisions. Even individuals with minimal technical knowledge will find it easy to set up the smart plugs and devices, without needing technical support. With interactive charts and real-time data at their fingertips, PowerEye offers an engaging, intuitive experience, making energy efficiency achievable for everyone.

Existing energy recommendation systems often focus on one specific aspect of energy optimization, such as reducing peak consumption, providing switch-on/off recommendations, or minimizing electricity costs. These systems tend to rely on single methods like collaborative filtering or rule-based approaches, which limit their effectiveness by not offering comprehensive solutions.

In contrast, PowerEye's EPR further enhances this by using a hybrid approach to energy optimization. It helps users set cost-saving goals, recommends peak-time savings, detects phantom power modes, and identifies excessive energy usage, addressing multiple aspects of energy efficiency in one solution.

B. System Architecture

The proposed home energy management system, PowerEye, has a modular architecture with system functionalities distributed among different modules or components, as shown in Fig. 1. Each module interacts with the others using specified protocols and APIs.

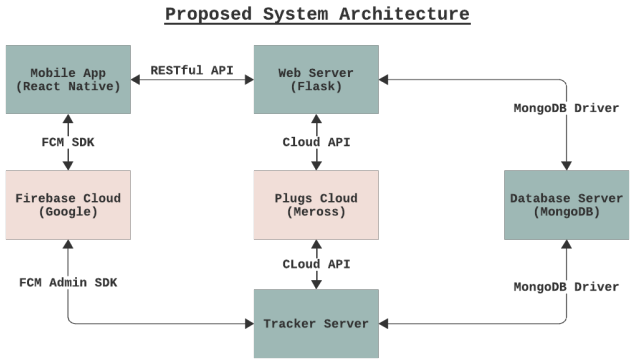


Fig. 1. System architecture design diagram.

The proposed PowerEye system is broken down into the following components:

1) Database server: This server stores and manages data such as user profiles, appliance information, and energy consumption records. MongoDB is utilized for its flexibility and scalability [17], making it ideal for efficiently handling time-series and semi-structured IoT data.

2) Plugs cloud (meross) [18]: The Plugs Cloud component acts as a bridge between the application and Meross smart plugs. It enables communication through an unofficial API provided by Alberto Geniola [19], enabling real-time information retrieval and remote control.

3) Web server: This server uses the Flask framework to handle HTTP requests via RESTful APIs, manage user authentication, and serve as the primary point of interaction between users and the application [20]. Flask [20] was selected for its lightweight nature and ease of integration with data-driven services.

4) Firebase cloud [21]: Firebase Cloud Messaging (FCM) is responsible for sending push notifications to mobile devices. It acts as a communication bridge between the Tracker Server and the mobile application. It ensures that users are promptly informed of critical events, such as reaching energy goals or receiving recommendations.

5) Tracker server: This is the backbone of the system, managing power data from Meross Cloud [18], storing device details in MongoDB [17], tracking events like energy goals, and delivering personalized recommendations and push notifications through FCM [21]. It centralizes backend operations, processes device data, and offers valuable insights.

6) Mobile application: The React Native-based mobile app offers a platform-independent interface for smart plug control [22]. Users can view device status, consumption data, and recommendations, and control devices remotely. React Native streamlines cross-platform app development with a single codebase, saving time and ensuring a consistent experience on Android and iOS.

C. Energy Personalized Recommender Design

In our proposed recommender system, we focus on four aspects of energy optimization: energy goal, peak time, phantom mode, and energy baseline.

1) Energy goal: Users can set monthly energy consumption goals based on cost. The system then calculates the energy equivalent to the specified cost. Users are then notified when their energy consumption hits specific percentage thresholds, ranging from 25% to 200%.

2) Peak time: Avoiding appliance usage during peak hours lowers emissions, conserves resources, cuts infrastructure costs, and supports renewable energy [23]. We will advise users to postpone appliance use during peak times.

3) Phantom mode: This refers to appliances consuming electricity even when they are turned off but still plugged in [24]. This happens because they stay in standby mode to respond quickly when needed. In this scenario, we will advise users to turn off appliances in phantom mode.

4) Energy baseline: Our system provides users with insights into their energy consumption patterns by sending alerts when they exceed their daily baseline. Each appliance in the system is equipped with its own model, allowing it to forecast energy consumption for the upcoming hour.

Based on the above aspects, we can conclude that our EPR employs a hybrid approach. It can be considered context-based as it uses contextual factors such as time, appliance status, power, and energy readings. Additionally, it qualifies as knowledge-based because it relies on user preferences and predefined rules for energy goals and peak times. Finally, it has content-based elements as it utilizes historical user energy patterns to predict future usage and issue alerts for excessive consumption.

D. Recommendation Process

The Tracker Server collects power readings from each Meross smart plug at one-minute intervals [18]. The EPR is designed using a modular architecture, in which the four optimization modules (energy goal tracking, peak-time shifting, phantom energy detection, and baseline forecasting) operate as independent components. Each module accesses the required input data from a shared data layer and does not rely on the outputs of other modules.

As the Tracker Server gathers data, it invokes the relevant EPR modules independently based on predefined conditions and scheduling logic, enabling the system to generate personalized recommendations for the user. The Tracker Server then sends these recommendations to the user's mobile application via Google Firebase [21]. This process is illustrated in Fig. 2.

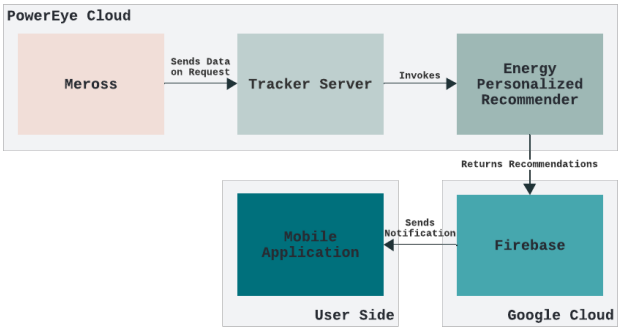


Fig. 2. Recommendation process.

The data used include user information, appliance details, power readings, and energy consumption. To calculate daily energy consumption, Eq. 1 is applied:

$$E(kWh) = \sum_{i=1}^{n} P_i \Delta t = \sum_{i=1}^{24\times 60} \frac{P_i}{1000} \times \frac{1}{60} \qquad (1)$$

Energy consumption is calculated by converting power from watts to kilowatts and multiplying it by the time interval, which is set at one minute (1/60 hour). The daily energy consumption values are then aggregated and stored with the unit kWh.

E. Energy Goal Algorithm

Users can set a monthly cost goal for their energy consumption. In Saudi Arabia, the energy cost is calculated by multiplying the energy consumption by the Residential Consumption Tariff (RCT), as specified on the official website [25]. The energy goal can then be calculated by rearranging Eq. 2 as shown in Eq. 3.

$$Cost(SAR) = E \times RCT = E \times \frac{0.8}{100} \tag{2}$$

$$E(kWh) = Cost \times \frac{100}{0.8} \tag{3}$$

The `check_goal()` function, provided by the EPR module, computes the percentage of the user's energy goal that has been reached. This percentage is determined by dividing the current month's energy consumption by the user's set goal, rounding down to the nearest quarter, and ignoring any quarter greater than 2. The result is then converted into a percentage out of 100. Upon reaching each specified threshold, the server sends a notification. Flags are used to prevent multiple notifications for the same percentage within the current month. The steps for this process are provided in Algorithm 1:

Algorithm 1 Check Goal

```
Input: month_energy (float), goal (float)
Output: int

function CHECK_GOAL(month_energy, goal)
    percentage ← month_energy / goal
    rounded ← ⌊ percentage * 4 ⌋ / 4
    if rounded ≤ 2 then
        return int(rounded * 100)
    else
        return 0
    end if
end function
```

F. Peak Time Algorithm

The `check_peak()` function, provided by the EPR module, uses straightforward logic. It checks whether the appliance is turned on during peak times and whether its energy type is categorized as shiftable. For more details on appliance types, refer to Appendix A. In Saudi Arabia, peak time is recognized by the government as 1–5 p.m., as stated on the government website [26]. If the function returns true, the server sends a notification. Flags are used to prevent additional notifications for the same appliance during the current day. The steps for this process are provided in Algorithm 2.

G. Phantom Mode Algorithm

Phantom appliances can exist in three distinct states: on, off, and phantom. The primary objective of this algorithm is to identify the phantom state. This is achieved by clustering appliance power data into two groups: `on_cluster` and `phantom_cluster`. The off state is intentionally included within the `phantom_cluster`, as phantom consumption typically occurs at very low power levels close to zero. Distinguishing between off and phantom states is then performed using the appliance status (ON/OFF), enabling accurate separation without requiring additional clustering complexity.

Algorithm 2 Check Peak

```
Input: cur_hour (int), status (boolean), e_type (EType), types (list)
Output: boolean

PEAK_START ← 13
PEAK_END ← 17
function CHECK_PEAK(cur_hour, status, e_type, types)
    is_peak ← PEAK_START ≤ cur_hour < PEAK_END
    return is_peak and status and e_type in types
end function
```

The dataset is extracted from the PowerEye database, containing timestamped appliance-level power readings. Only power values are used for clustering. The preprocessing steps are as follows:

- Null Values: These can be dropped as there is no need for imputation in this context.
- Dataset Size: A sample size of 1,000 readings was selected as a practical balance between data collection requirements and computational cost. The resulting clustering performance is reported through appliance-level silhouette scores (Table IV); sensitivity to sample size remains to be evaluated.
- Duplicate Values: Rather than being removed, duplicates are assigned weights to preserve their frequency information while avoiding bias in clustering.

For clustering, k-means with $k = 2$ is employed [27]. This choice is motivated by the univariate nature of the data (power readings only) and the expectation of two dominant consumption patterns: active and near-zero power. Compared to more complex methods such as DBSCAN or Gaussian mixture models, k-means provides a lightweight and deterministic solution suitable for real-time deployment [28]. Density-based methods like DBSCAN require sensitive parameter tuning and are less stable in one-dimensional data, while probabilistic models such as GMM introduce unnecessary complexity without significant performance gains for this task. Similar approaches have been effectively used in prior work [4]. A publicly available Jupyter notebook provides more detailed information [29]. One model is created for each appliance and stored on PowerEye's server, where it is used within the `check_phantom()` function.

The `check_phantom()` function, provided by the EPR module, first checks whether a cluster model already exists for the appliance. If the appliance is turned on, it uses the model to predict the cluster of the current power reading. The algorithm compares the predicted cluster with the cluster associated with a power reading of 0. If they match, it indicates that the appliance is in phantom mode, and the function returns True. When the function returns True, the server sends a notification. In cases where the clusters are not well separated, flags are used to avoid sending excessive notifications in a short period. The steps for this process are detailed in Algorithm 3.

Algorithm 3 Check Phantom

```
Input: model (clusterModel), power (float), status (boolean)
Output: boolean

function CHECK_PHANTOM(model, power, status)
    if model and status then
        predicted_labels ← model.predict([0, power])
        return predicted_labels[0] == predicted_labels[1]
    end if
    return False
end function
```

H. Energy Baseline Algorithm

The final EPR step alerts users of potential excessive usage by forecasting appliance-level energy consumption. If actual consumption deviates from predictions, users are notified that their usage exceeds expected levels. This approach helps users monitor and reduce energy usage. Machine learning models predict hourly consumption based on historical data, setting an expected energy limit. Users are notified if they exceed this limit.

To prevent data leakage and preserve temporal causality, all preprocessing and model evaluation steps strictly respect the chronological order of observations. The preprocessing pipeline is defined as follows:

1) Energy conversion and hourly resampling: Power readings are converted into energy using Eq. 1 with $n = 60$, then resampled hourly, generating more data for better prediction accuracy and quicker energy-saving decisions.

2) Null values: Rows with more than 30 null values (before resampling) are dropped.

3) Normalization: A standard scaler is used to handle large magnitude differences. The dataset is split 80/20 into training and test sets before normalization to keep the model blind to the test data.

4) Feature engineering: To transform the time series into a regression problem, we decompose the timestamp into features like hour, month, year, day, and whether it is a weekend or nighttime. We then add rolling statistics (e.g., average, standard deviation) to capture trends and variations. Lastly, we include seven lag features to balance predictive power and computational efficiency.

5) Dataset splitting: Half of the testing dataset is reallocated for validation, resulting in a new ratio of 80% for training, 10% for validation, and 10% for testing (0.8:0.1:0.1). This ensures that all training observations precede validation observations, and validation observations precede test observations, preventing future information from leaking into model training.

6) Feature selection: The most relevant features are identified by assessing their impact on the model's predictions. Using XGBoost with default hyperparameters [30], the top 15 features with the highest importance scores are selected to improve model performance and reduce overfitting.

For forecasting, the XGBoost model—short for eXtreme Gradient Boosting—is employed using time-series cross-validation with walk-forward validation to maintain the dataset's temporal structure [30]. This approach trains on historical data and validates subsequent points, avoiding look-ahead bias and providing realistic performance assessments. Hyperparameter tuning is essential since XGBoost has numerous hyperparameters. The tuning process followed Jain's approach [31], using Mango Tuner instead of grid search cross-validation. Mango Tuner is a state-of-the-art gradient-free optimizer that offers faster performance and greater flexibility in setting objective functions [32]. The objective function employs time-series cross-validation (walk-forward) and aims to minimize mean squared error (MSE).

After finding the optimal hyperparameters, the model is trained using these parameters and then utilized to predict the energy consumption for the following hour, which serves as a baseline. An actual value is considered abnormal if it significantly deviates from this baseline. A normalized relative MAE score is used, where values exceeding a threshold of 0.75 are flagged as excessive usage [33]. This threshold was chosen as a conservative operating point to balance sensitivity and false-positive rates in practical deployment. Lower thresholds (e.g., 0.6–0.7) were observed to increase false alerts due to short-term fluctuations and sensor noise, while higher thresholds (e.g., above 0.8) reduced sensitivity to meaningful deviations. The score is calculated by taking the absolute difference between the predicted and actual values, dividing it by the mean absolute error (MAE) [33], and then normalizing the result to compare against the fixed threshold. The formulas are presented in Eq. 4, 5, and 6.

$$MAE = \frac{\sum_{i=1}^{n} |Predicted_i - Actual_i|}{n} \tag{4}$$

$$Score_i = \frac{|Predicted_i - Actual_i|}{MAE_{total}} \tag{5}$$

$$NormalizedScore_i = \frac{Score_i - min(Score)}{max(Score) - min(Score)} \tag{6}$$

The optimal hyperparameters, along with the MAE, maximum score, and minimum score obtained during the tuning phase, are stored on PowerEye's server for later use in the `check_baseline()` function. This function, part of the EPR module, applies the same preprocessing steps described earlier to all power readings up to the current hour. It then splits the data to train the XGBoost model and tests it on the most recent reading. The function calculates the normalized relative MAE and returns True if the actual energy consumption exceeds the predicted value and the score surpasses the 0.75 threshold. If True, the server sends a notification. Flags are used to prevent sending multiple notifications for the same appliance within the same day. The process is outlined in Algorithm 4, with detailed steps available in our Jupyter notebook [29].

IV. PowerEye Simulation

The following section includes details about the deployment and experimental setup.

Algorithm 4 Check Baseline

Input: powers (time-series), mae (float), max_score (float), min_score (float), params (dict)
Output: boolean

function CHECK_BASELINE(powers, mae, max_score, min_score, params)
    count null values for each hour
    drop any row with more than 30 null values
    df["energy"] ← (powers/1000) * (1/60)
    df ← df.resample("H").sum()
    df ← normalize(df)
    df ← feature_engineering(df)

    y ← df["energy"]
    X ← df.drop("energy")
    y_train, y_test ← y[:-1], y[-1]
    X_train, X_test ← X[:-1], X[-1]

    model ← XGBRegressor(params)
    model.fit(X_train, y_train)
    pred ← model.predict(X_test)

    r_mae ← abs(pred - y_test) / mae
    nr_mae ← (r_mae - min_score) / (max_score - min_score)
    return y_test > pred **and** nr_mae > 0.75
end function

A. Proposed System Deployment

The deployment of PowerEye involved configuring both hardware and software systems. All deployment configuration details for the proposed system are summarized in Table II.

TABLE II. POWEREYE DEPLOYMENT CONFIGURATIONS

Component	Configuration
Database	MongoDB Atlas on AWS (Bahrain Region), M0 Sandbox cluster, 512 MB storage, shared RAM
Firebase (FCM)	Configured for iOS and Android with secure authentication keys, APN for iOS, SHA certificate for Android
Web Server	AWS EC2 t3.micro instance, Ubuntu Server 22.04 LTS, 30 GB SSD storage, 2 vCPUs, 1 GiB RAM
Tracker Server	Orange Pi 5 SBC, Rockchip RK3588S 8-core processor, 8 GB LPDDR4 RAM, 32 GB microSD storage, Ubuntu 22.04.2 LTS
Mobile Application	React Native with Expo, iOS and Android deployment, Apple Distribution certificates, ad hoc provisioning profiles for push notifications and internal distribution

B. Experimental Setup

A total of 23 commercial smart plugs were purchased and distributed among three participants with different characteristics, as detailed in Table III.

Data collection for the HEMS lasted for 111 days, initially using temporary cloud-hosted code. Later, the Tracker Server took over the data collection process. The participants installed the PowerEye application on their mobile devices, enabling them to access their accounts and monitor their data. The mobile application experiment was limited to a single day due to time and resource constraints.

TABLE III. EXPERIMENTAL PARTICIPANT INFORMATION

User	Rooms	Family Members	Smart Plugs	Appliance Types
User 1	10	9	8	4 Phantom, 4 None
User 2	8	7	6	1 Phantom, 5 None
User 3	8	5	9	4 Phantom, 5 None

V. RESULTS

The following section discusses the evaluation of the experiments using phantom clustering and baseline forecasting accuracy, user experience, and comparative analysis.

A. Phantom Clustering Accuracy

Scatterplots for the power clusters of all phantom appliances across users are presented in Fig. 3. The quality of the k-means clustering models is evaluated using the silhouette score [34], which quantifies the similarity of data points within their cluster (cohesion) relative to other clusters (separation). Silhouette scores range from -1 to 1, with higher scores indicating better clustering. Table IV provides the silhouette scores, rounded to three decimal places, for all 2-means models of phantom appliances.

TABLE IV. APPLIANCE SILHOUETTE SCORES

User 1			
office_strip_10: 0.572	tv_10: 0.678	tv_11: 0.788	charger_10: 0.637
User 2			
charger_20: 0.503			
User 3			
speaker_30: 0.915	tv_30: 0.609	charger_30: 0.478	office_strip_30: 0.954

B. Baseline Forecasting Accuracy

Line plots illustrating the energy consumption of all appliances across users are presented in Fig. 4. The blue line represents the actual energy consumption, the red dashed line indicates the predicted energy consumption, and the black dots highlight the actual values that significantly deviate from the predicted ones. Since this is a regression problem, Root mean squared error (RMSE) and MAE were used to evaluate the XGBoost models for each appliance. All scores are presented in Table V, rounded to three decimal places. Appliances that did not meet the requirements for the XGBoost model were omitted from the table.

C. Energy Consumption Analysis

We conducted a one-day experiment with three existing users via the PowerEye app (Fig. 5). Table VI compares measured energy consumption on the experiment day with the previous day.

D. User Experience

To evaluate user satisfaction with the PowerEye application, we created a survey that includes several questions

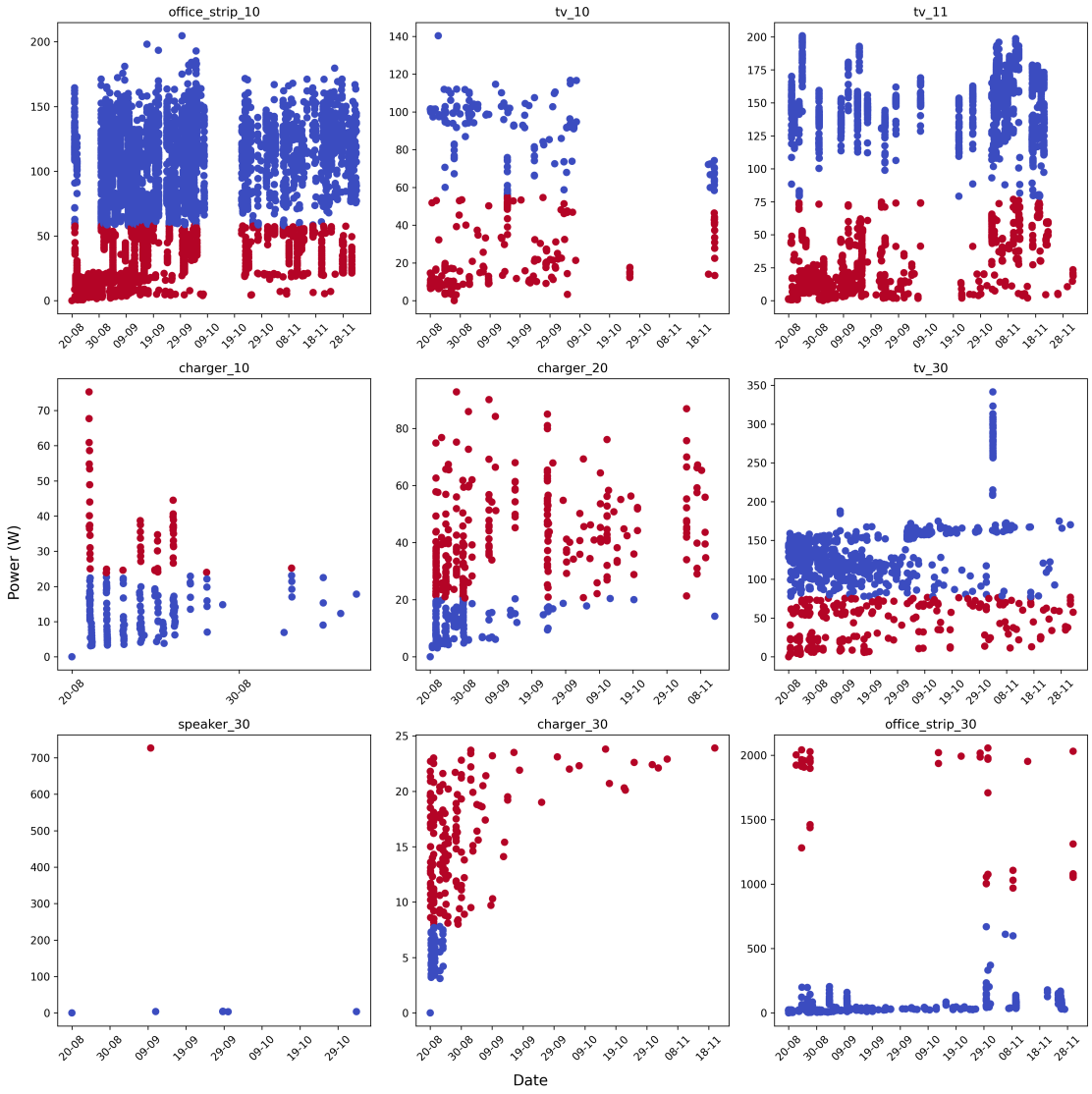


Fig. 3. Phantom appliance power clusters using k-means clustering models.

about their experience. The survey specifically assesses the relevance and helpfulness of the recommendations, as well as how effectively the application aids users in optimizing their energy consumption. Below are the questions related to the EPR and their respective responses, as shown in Fig. 6:

1) Peak time notifications: If you received peak time notifications, did the appliance mentioned in the notification match one whose use you could postpone? Did this notification help you optimize your energy consumption?

2) Goal notifications: If you received a goal notification, did the percentage mentioned match the percentage displayed in the app? Did this notification assist you in understanding your energy consumption?

3) Phantom mode notifications: If you received a phantom mode notification, was your device actually in standby mode? Did this notification assist you in eliminating energy wastage?

4) Energy baseline notifications: If you received an energy baseline notification, did you think the appliance mentioned in the notification was used excessively during that hour? Did it help you better understand your energy consumption?

VI. Discussion

A. Phantom Clustering

As shown in Fig. 3, some of the scatterplots exhibit high separation between clusters, while others do not. We can establish a threshold of 0.5 for our models. In practice, average silhouette values between 0.5 and 0.7 are commonly interpreted as reflecting moderate to good clustering structure, providing a balanced criterion for identifying meaningful clusters while avoiding over-sensitivity to noise. We can also introduce a flag to avoid sending too many notifications in a short period.

There are two cases when the model predicts incorrectly:

- False Positive: The model incorrectly classifies an on_state as phantom, leading to potentially annoying notifications. User feedback can help mitigate this.

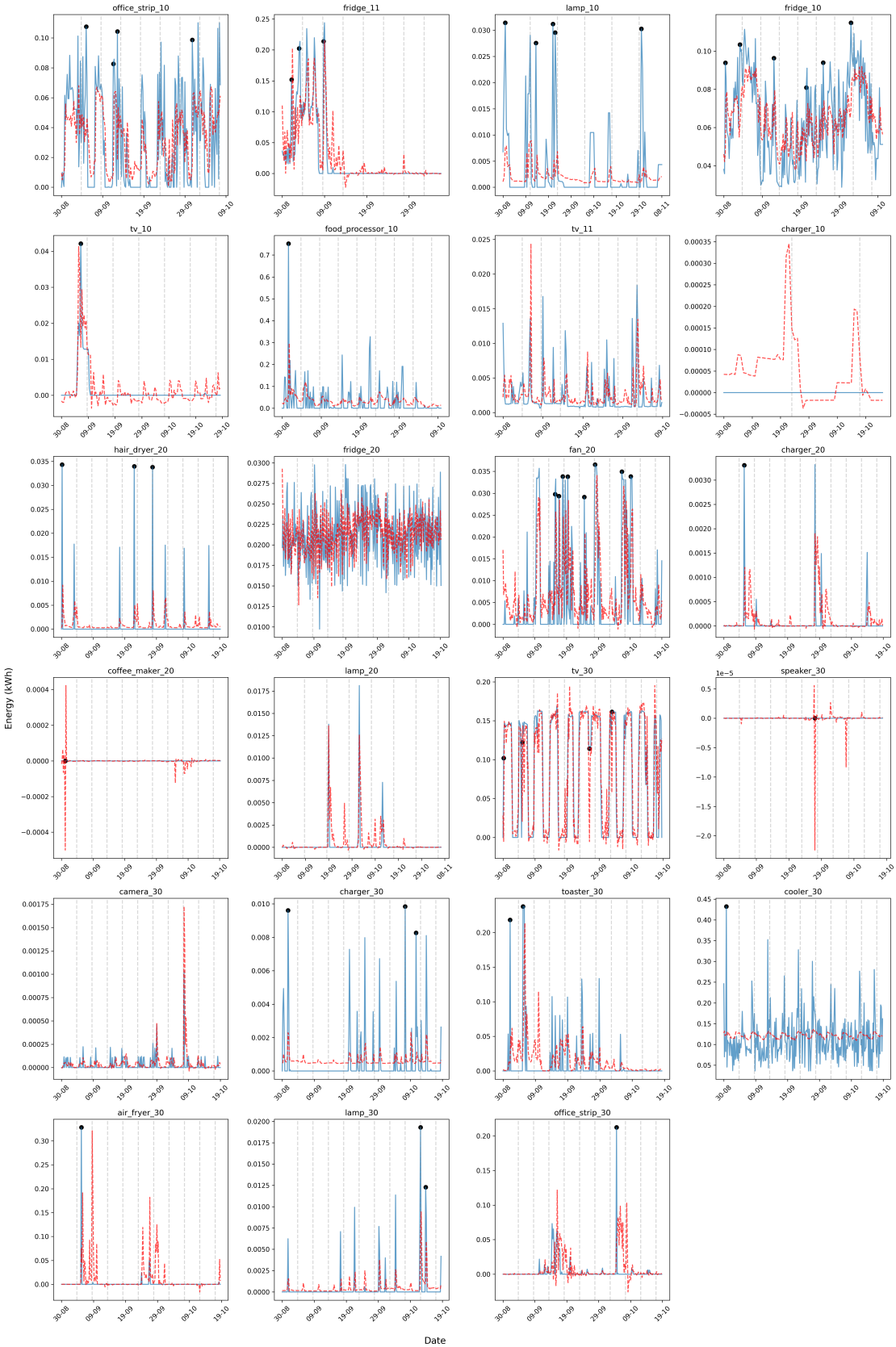


Fig. 4. Actual vs. predicted appliance energy consumption.

TABLE V. APPLIANCE MAE AND RMSE SCORES

User 1					
office_strip_10		**fridge_11**		**lamp_10**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.788	1.024	0.300	0.603	0.834	1.526
tv_10		**food_processor_10**		**tv_11**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.060	0.112	0.492	0.904	0.084	0.147
fridge_10		**charger_10**			
MAE	RMSE	MAE	RMSE		
0.717	0.849	0.017	0.026		
User 2					
hair_dryer_20		**fridge_20**		**fan_20**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.371	1.086	0.510	0.647	0.437	0.617
coffee_maker_20		**lamp_20**		**charger_20**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.002	0.011	0.068	0.218	0.055	0.151
User 3					
tv_30		**speaker_30**		**camera_30**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.249	0.432	0.000	0.003	0.253	0.639
toaster_30		**cooler_30**		**air_fryer_30**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.647	1.239	0.706	0.953	0.266	0.906
office_strip_30		**charger_30**		**lamp_30**	
MAE	RMSE	MAE	RMSE	MAE	RMSE
0.389	0.948	0.478	0.824	0.227	0.589

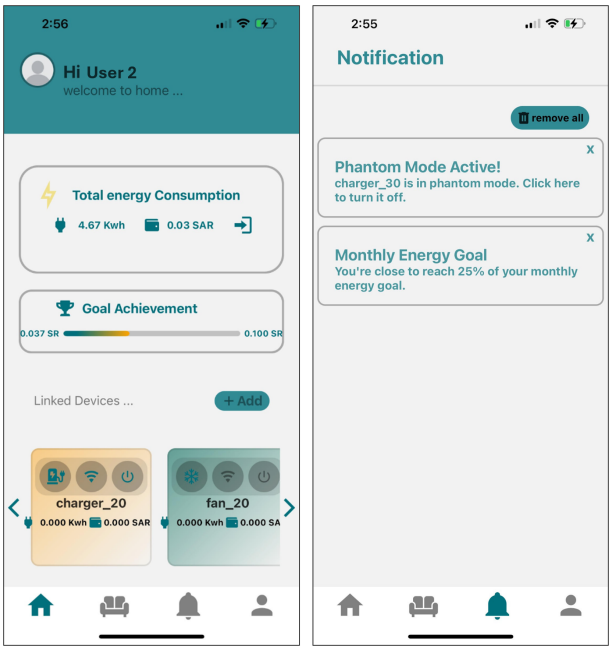


Fig. 5. PowerEye mobile application interfaces.

TABLE VI. ENERGY CONSUMPTION BEFORE AND AFTER THE EXPERIMENT

	Before (kWh)	After (kWh)	Change (%)
User 1	0.802	0.796	−0.75
User 2	0.720	0.682	−5.28
User 3	4.374	5.582	+27.62
Total	5.896	7.060	+19.74

Change = $100 \times (\text{After} - \text{Before})/\text{Before}$; negative values indicate reductions.

- False Negative: The model incorrectly classifies a phantom as on_state, missing a notification. This reduces energy optimization but has a minimal impact.

B. Baseline Forecasting Accuracy

Forecasting accuracy varied across appliances (Table V). Several models achieved low errors on the standardized consumption scale, including RMSE values of 0.147 for `tv_11` and 0.218 for `lamp_20`, while others showed larger deviations. These results characterize appliance-level prediction accuracy; comparison with a historical-average or persistence predictor is needed to establish the benefit of XGBoost over simpler forecasts.

Forecasting errors can affect excessive-use detection in two ways:

- False Positive: The model wrongly predicts excessive energy use. We can reduce this by limiting notifications to once a day.
- False Negative: The model fails to detect excessive energy use. This missed notification impacts optimization minimally.

Due to the experiment's short duration and resource constraints, we were unable to evaluate additional models. However, more advanced time-series models such as long short-term memory (LSTM) networks or autoregressive integrated moving average (ARIMA) could be considered in future work, as they are designed to capture temporal dependencies and may improve forecasting accuracy. These models, however, require higher computational resources and careful tuning to avoid overfitting, particularly when data are limited.

C. Energy Consumption Analysis

Table VI shows mixed outcomes: two participants reduced consumption, but the third participant's increase raised total consumption from 5.896 to 7.060 kWh (19.74%). Thus, the experiment did not demonstrate an overall energy reduction. The single-day intervention, small sample, and absence of a control group prevent attributing these changes to EPR rather than ordinary day-to-day variation. Longer controlled trials are needed to evaluate sustained energy savings.

D. User Experience

The majority of users reported satisfaction with PowerEye's EPR notifications (Fig. 6). While precision/recall metrics and user engagement rates are commonly used to evaluate recommendation effectiveness, the limited number of participants

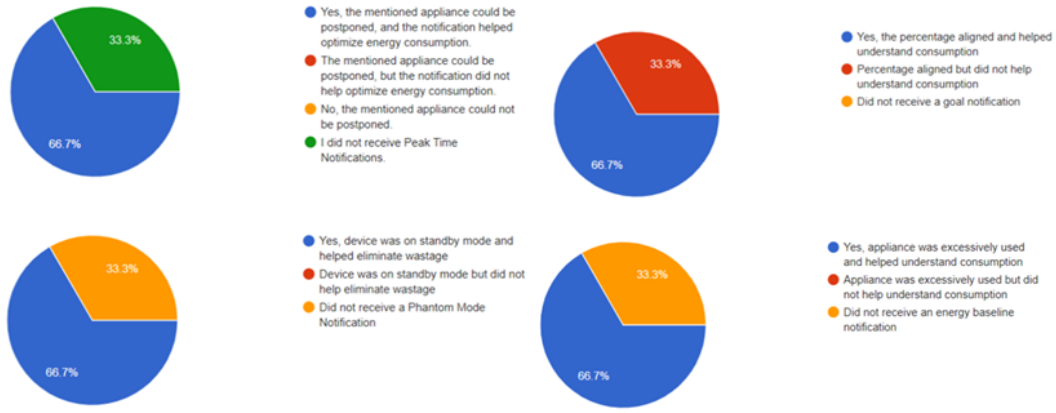


Fig. 6. User experience evaluation responses.

TABLE VII. EXISTING ENERGY RECOMMENDATION SYSTEMS VS. THE PROPOSED EPR

	Type	Multi-agent system	Uses price data	User in control	Peak time savings	Anomaly detection	Phantom mode detection
[10]	Hybrid	✓	✓	✓	✓		
[4]	Hybrid			✓	✓	✓	
[11]	Case-based	✓	✓		✓		
[14]	Collaborative		✓	✓	✓		
[5]	Hybrid			✓			
[15]	Content-based		✓	✓	✓		
[16]	Content-based			✓		✓	
Our EPR	Hybrid			✓	✓	✓	✓

in this study prevents statistically meaningful computation of such measures. Nevertheless, the observed user feedback suggests that the notifications were effective in supporting users' energy management and optimization efforts. These findings should be interpreted cautiously, as they are based on a small sample size and short-term deployment. A long-term evaluation involving a larger and more diverse user population is required to rigorously assess the system's effectiveness and its sustained impact on energy-saving behavior.

E. Comparative Analysis

Table VII presents a comparative analysis between existing energy recommendation systems and the proposed Energy Personalized Recommender (EPR). The comparison highlights key characteristics such as recommender type, use of price data, level of user control, and supported functionalities including peak-time optimization, anomaly detection, and phantom load detection.

Existing systems, including REHAB-C [5] and other hybrid or single-paradigm approaches, typically focus on a subset of energy management functionalities. For instance, some systems emphasize price-aware scheduling or peak-time reduction, while others address anomaly detection or provide user feedback. However, these approaches do not simultaneously integrate 1) appliance-level monitoring, 2) real-time adaptive recommendations, 3) active user interaction, and 4) multi-objective optimization within a unified framework. As a result, they remain limited either in personalization, responsiveness, or scope of optimization.

The proposed EPR integrates four complementary modules: energy goal tracking, peak-time shifting, phantom load detection, and baseline forecasting. This integration is intended to support multiple energy-management objectives within one system; its comparative effect on energy savings remains to be evaluated.

Eirinaki et al. [4] report energy reductions of 2–17%, while REHAB-C [5] reports reductions of up to 50%. In PowerEye, consumption decreased by 0.75% and 5.28% for two participants but increased by 27.62% for the third, yielding an aggregate increase of 19.74% (Table VI). This preliminary evaluation therefore did not demonstrate an overall energy reduction. Differences in study settings and evaluation methods preclude a direct performance ranking. Table VII describes functional coverage, not superior energy-saving effectiveness.

F. Assumptions and Limitations

The PowerEye project was conducted within the scope of a research-based laboratory prototype, which imposed time, hardware, and deployment constraints that influenced the study outcomes. These constraints limited the duration of data collection and the number of participating users. In particular, the evaluation relied on a small user cohort and a finite dataset, which restricts the statistical generalizability of the reported results.

Hardware resource limitations also constrained real-time processing under higher data volumes and precluded large-scale deployment. Additionally, certain edge cases, such as

irregular appliance usage patterns or atypical household behaviors, were underrepresented, which may affect the robustness of the EPR in more diverse real-world scenarios.

Nevertheless, the primary objective of this study was to validate the functional feasibility and effectiveness of the proposed EPR rather than to provide population-level conclusions. The system is designed as a modular and user-independent framework, and the employed algorithms are computationally scalable. Consequently, the proposed approach is suitable for extension to larger user bases and longer deployment periods.

VII. Conclusions and Future Work

This paper explored PowerEye's EPR for personalized energy-management recommendations. Preliminary results suggest the feasibility of its energy goal, peak time, phantom mode, and baseline algorithms, and participating users reported satisfaction with its notifications. However, the single-day deployment involving three participants did not demonstrate an overall energy reduction or establish sustained user engagement. Longer controlled trials with a larger and more diverse user base are needed to assess prediction accuracy, recommendation effectiveness, and measurable energy savings.

Acknowledgments

The authors would like to thank the software engineering team members, Maya Emar, Nour Al Akhras, Joody Koujan, Ward Najjar, Raneem Balharith, Sarah Alshagawi, and Renad Almahmoud, for their valuable assistance in developing the system. We also gratefully acknowledge Prince Sultan University for funding this research.

Declarations

Ethical Approval

This study was approved by the Institutional Review Board of Prince Sultan University (Approval No. PSU-IRB-2025-09-0245).

Consent to Participate

All participants provided informed consent prior to their participation in the study.

Funding

This study was funded by the College of Computers and Information Science at Prince Sultan University.

Competing Interests

The authors declare no competing interests.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Code Availability

The code for this study is publicly available on GitHub at the following link: https://github.com/rem2718/PowerEye-backend.

Generative AI Usage Disclosure

The authors used generative AI tools solely for language refinement and readability, with all AI-assisted content reviewed and verified by the authors.

Author Contributions

Conceptualization, R.H. and A.A.; methodology, R.H. and A.A.; software, R.H.; validation, R.H.; formal analysis, R.H.; investigation, R.H.; resources, R.H.; data curation, R.H.; writing—original draft preparation, R.H.; writing—review and editing, A.A., R.H., and S.A.M.; visualization, R.H.; supervision, A.A., S.A.M., and L.A.; project administration, S.A.M., and L.A.; funding acquisition, S.A.M. All authors have read and agreed to the published version of the manuscript.

References

[1] Mahapatra B, Nayyar A (2019) Home energy management system (hems): Concept, architecture, infrastructure, challenges and energy management schemes. Energy Systems https://doi.org/10.1007/s12667-019-00364-w, URL https://link.springer.com/article/10.1007/s12667-019-00364-w

[2] Himeur Y, Alsalemi A, Al-Kababji A, et al (2021) A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects. Information Fusion 72:1–21. https://doi.org/10.1016/j.inffus.2021.02.002, URL https://www.sciencedirect.com/science/article/pii/S1566253521000178

[3] Sarmas E, Marinakis V, Doukas H (2025) Meta-Learning Approaches for Assessing Energy Efficiency Investments in Buildings, Springer Nature Switzerland, Cham, pp 151–176. https://doi.org/10.1007/978-3-031-85209-1_6, URL https://doi.org/10.1007/978-3-031-85209-1_6

[4] Eirinaki M, Varlamis I, Dahihande J, et al (2022) Real-time recommendations for energy-efficient appliance usage in households. Frontiers in Big Data 5. https://doi.org/10.3389/fdata.2022.972206, URL https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2022.972206

[5] Sardianos C, Varlamis I, Dimitrakopoulos G, et al (2020) Rehabc: Recommendations for energy habits change. Future Generation Computer Systems 112:394–407. https://doi.org/10.1016/j.future.2020.05.041, URL https://www.sciencedirect.com/science/article/pii/S0167739X19317327

[6] Badr AA, Saafan MM, Abdelsalam MM, et al (2023) Novel variants of grasshopper optimization algorithm to solve numerical problems and demand side management in smart grids. Artificial Intelligence Review 56:10679–10732. https://doi.org/10.1007/s10462-023-10431-5, URL https://doi.org/10.1007/s10462-023-10431-5

[7] AbdElkader AG, ZainEldin H, Saafan MM (2024) Optimizing wind power forecasting with rnn-lstm models through grid search cross-validation. Sustainable Computing: Informatics and Systems 45:101054. https://doi.org/10.1016/j.suscom.2024.101054, URL https://doi.org/10.1016/j.suscom.2024.101054

[8] Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: State of the art and trends. In: Ricci F, Rokach L, Shapira B, et al (eds) Recommender Systems Handbook. Springer, Boston MA, p 73–105

[9] Herlocker JL, Konstan JA, Terveen LG, et al (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5—-53. https://doi.org/10.1145/963770.963772, URL https://doi.org/10.1145/963770.963772

[10] Jiménez-Bravo DM, Pérez-Marcos J, H. De la Iglesia D, et al (2019) Multi-agent recommendation system for electrical energy optimization and cost saving in smart homes. Energies 12(7). https://doi.org/10.3390/en12071317, URL https://www.mdpi.com/1996-1073/12/7/1317

[11] González-Briones A, Prieto J, De La Prieta F, et al (2018) Energy optimization using a case-based reasoning strategy. Sensors 18(3). https://doi.org/10.3390/s18030865, URL https://www.mdpi.com/1424-8220/18/3/865

[12] Villegas NM, Sánchez C, Díaz-Cely J, et al (2018) Characterizing context-aware recommender systems: A systematic literature review. Knowledge-Based Systems 140:173–200. https://doi.org/10.1016/j.knosys.2017.11.003, URL https://www.sciencedirect.com/science/article/pii/S0950705117305075

[13] Wooldridge M, Jennings NR, Kinny D (2000) The gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems 3(3):285–312. https://doi.org/10.1023/A:1010071910869, URL https://doi.org/10.1023/A:1010071910869

[14] Zhang Y, Meng K, Kong W, et al (2019) Collaborative filtering-based electricity plan recommender system. IEEE Transactions on Industrial Informatics 15(3):1393–1404. https://doi.org/10.1109/TII.2018.2856842

[15] Zhang Y, Meng K, Xu D, et al (2016) Recommending electricity plans: A data-driven method. In: 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp 668–673, https://doi.org/10.1109/SmartGridComm.2016.7778838

[16] Rodríguez-Cuenca F, Sánchez-Úbeda EF, Portela J, et al (2023) Probability-density-based energy-saving recommendations for household refrigerating appliances. Engineering Proceedings 39(1). https://doi.org/10.3390/engproc2023039043, URL https://www.mdpi.com/2673-4591/39/1/43

[17] MongoDB (2024) Mongodb atlas: Multi-cloud developer data platform. Available online: https://www.mongodb.com/atlas (accessed on 1 March 2025), URL https://www.mongodb.com/atlas

[18] Meross (2024) Meross. Available online: https://www.meross.com (accessed on 1 March 2025), URL https://www.meross.com

[19] Geniola A (2020) Merossiot api documentation. Available online: https://albertogeniola.github.io/MerossIot (accessed on 1 March 2025), URL https://albertogeniola.github.io/MerossIot

[20] Pallets Projects (2024) Flask, a lightweight wsgi web application framework. Available online: https://flask.palletsprojects.com/en/stable (accessed on 1 March 2025), URL https://flask.palletsprojects.com/en/stable

[21] Google (2024) Firebase cloud messaging (fcm). Available online: https://firebase.google.com/docs/cloud-messaging (accessed on 1 March 2025), URL https://firebase.google.com/docs/cloud-messaging

[22] Facebook (2024) React native. Available online: https://reactnative.dev (accessed on 1 March 2025), URL https://reactnative.dev

[23] Benetti G, Caprino D, Della Vedova ML, et al (2016) Electric load management approaches for peak load reduction: A systematic literature review and state of the art. Sustainable Cities and Society 20:124–141. https://doi.org/10.1016/j.scs.2015.05.002, URL https://www.sciencedirect.com/science/article/pii/S2210670715000529

[24] Dawson E, Potter J, Meier S (2012) Are phantom loads haunting your energy bill? Available online: http://pages.uoregon.edu/hof/W09HOF/19Phantom_ppr.pdf (accessed on 1 March 2025), URL http://pages.uoregon.edu/hof/W09HOF/19Phantom_ppr.pdf

[25] Saudi Electricity Company (2024a) Consumption tariffs. Available online: https://www.se.com.sa/en/Ourservices/ColumnC/Bills-and-Consumption/ConsumptionTariffs (accessed on 1 March 2025), URL https://www.se.com.sa/en/Ourservices/ColumnC/Bills-and-Consumption/ConsumptionTariffs

[26] Saudi Electricity Company (2024b) Consumption tariffs, consumption rationalization. Available online: https://www.se.com.sa/en/Electricity-Information/Consumption-Rationalization/Industrial-Sector (accessed on 1 March 2025), URL https://www.se.com.sa/en/Electricity-Information/Consumption-Rationalization/Industrial-Sector

[27] Wu J (2012) Advances in K-means Clustering: A Data Mining Thinking, 1st edn. Springer Theses, Springer Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-29807-3, URL https://doi.org/10.1007/978-3-642-29807-3, springer-Verlag Berlin Heidelberg

[28] Bishop CM (2006) Pattern Recognition and Machine Learning. Springer

[29] Hejazi R (2024) Energy personalized recommender kaggle notebook. Available online: https://www.kaggle.com/code/reemhejazi/energy-personalized-recommender (accessed on 1 March 2025), URL https://www.kaggle.com/code/reemhejazi/energy-personalized-recommender

[30] Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, KDD '16, p 785–794, https://doi.org/10.1145/2939672.2939785, URL https://doi.org/10.1145/2939672.2939785

[31] Jain A (2024) Xgboost parameters tuning: A complete guide with python codes. Available online: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python (accessed on 1 March 2025), URL https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python

[32] ARM Software (2020) Mango: A parallel hyperparameter tuning library. Available online: https://github.com/ARM-software/mango (accessed on 1 March 2025), URL https://github.com/ARM-software/mango

[33] Zhang J, Zhang H, Ding S, et al (2021) Power consumption predicting and anomaly detection based on transformer and k-means. Frontiers in Energy Research 9. https://doi.org/10.3389/fenrg.2021.779587, URL https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2021.779587

[34] Rousseeuw PJ (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7, URL https://www.sciencedirect.com/science/article/pii/0377042787901257

APPENDIX A

In this research, home appliances were categorized into three types based on their common energy patterns: Shiftable, Phantom, and None. Shiftable appliances are those whose usage can be adjusted or shifted by residents. Phantom appliances are those that consume power even when they appear to be off (Phantom Mode). The None category includes appliances that do not fit into either of the other two categories. It is important to note that all phantom appliances are considered shiftable, but not all shiftable appliances are phantom. The appliance types are demonstrated in Table VIII.

TABLE VIII. APPLIANCE TYPES

Type	Energy Type	Appliances
Coolers	None	Fridge, Freezer, Mini Fridge, Water Cooler, Fan
Lighting	None	Table Lamp, Desk Lamp, Floor Lamp, Mirror Light
Heaters	None	Microwave, Toaster, Stove, Kettle
Cookers/Makers	None	Coffee Maker, Popcorn Maker, Air Fryer, Slow Cooker
Blenders	None	Blender, Food Processor
No Category	None	Clock/Alexa, Hair Dryer, Camera
Washing Machines	Shiftable	Washer, Dishwasher, Clothes Dryer
No Category	Shiftable	Iron, Vacuum Cleaner, Air Purifier
Games	Phantom	PlayStation, Wii, Xbox
Displays	Phantom	PC, Monitor, Projector, TV
Audio Output	Phantom	Speaker, Radio
No Category	Phantom	Printer, Charger, Receiver, Sports Machine

Anatomia: An Interactive 3D Visualization Tool for Enhancing Medical Anatomy Education

Iranga Mudalige[1], Hiruna Jayasuriya[2], Yeshan Gunawardane[3], Hashen Cooray[4],
Imashi Dissanayake[5], Damitha Sandaruwan[6],Chamath Keppitiyagama[7],
Mahappuge Shasika Eranda Karunadasa[8], Tony Mahadevan[9]
University of Colombo School of Computing, Colombo 00700, Sri Lanka[1,2,3,4,5,6,7]
The Dudley Group NHS Foundation Trust, United Kingdom[8]
C3 & Pixel Paper Labs, Colombo 00300, Sri Lanka[9]

***Abstract*—Anatomy education continues to rely on cadaver dissection, which is constrained by donor shortages, costs, and limited hands-on access. Digital 3D anatomy tools offer alternatives, but most do not collaborate well across multiple users, keep upright content legible as viewers move around the table, or support hand gestures specific to anatomical interaction. This study presents Anatomia, a multi-touch tabletop application designed for around-the-table anatomy learning. It introduces a set of domain-specific hand gestures developed with medical students and instructors, a content-alignment mechanism that adjusts display orientation based on the user's position around the table, and a software-level conflict resolver that allows two users to share a single display without additional hardware. Anatomia was evaluated with medical and non-medical students under a structured two-phase protocol. Results indicate that familiar touchscreen gestures succeed on first attempts, anatomically novel gestures are learned within a single guided session, content-alignment tasks score near the ceiling, and concurrent multi-user interactions achieve a perfect task success rate with no participant rejections. Together, the findings indicate that gesture-driven collaborative 3D anatomy visualization on commodity touchscreen hardware is a viable complement to traditional dissection, especially where specialized anatomy visualization tables are not affordable.**

***Keywords*—3D anatomy visualization; anatomy education; gesture interaction; multi-touch interaction; content alignment; tabletop display; collaborative learning**

I. Introduction

In the field of medical science, understanding human anatomy is fundamental for clinical knowledge and practice. For many years, anatomy education has rested on a combination of cadaver dissection, demonstrations, anatomical models, and lectures [1]. Among these, cadaver dissection has been particularly significant. It supports both anatomical knowledge and the manual dexterity, tissue-handling skills, teamwork, and professionalism that medical students require [1], [2]. Despite these benefits, cadaver dissection alone cannot easily demonstrate functional neuroanatomy, complete skeletal relationships, or the fine vascular and neural structures encountered in microsurgery [3]. Practical limitations further constrain its use: donor shortages, increasing lab costs, reduced teaching time, and health concerns related to formaldehyde exposure [4], [5], [1], [2]. These constraints are particularly tough in areas where resources are limited. Typically, in a traditional dissection lesson in medical faculties in Sri Lanka, a group of 10-20 students are assigned to a single human cadaver to carry out the studies throughout the year. This may vary with the university location, as most of the outstation universities receive a few cadavers per year. These insights highlight that there is a need for an accessible, shared-surface visualisation tool that can support cadaver-based instruction without requiring any additional cadaveric resources.

Digital tools are increasingly used alongside these traditional methods rather than replacing them. Among the available digital approaches, three-dimensional (3D) visualization has drawn particular attention: the body is inherently three-dimensional, so the ability to explore structures from different angles, isolate systems, view cross-sections, and rotate, flip, and zoom aligns naturally with how anatomy is learned and applied. According to research, 3D visualization improves overall learning experiences, increases teaching efficiency, and can be more cost-effective compared to some traditional methods [6], [4], [2], [7].

Collaborative learning around a shared surface is not a supplementary feature in anatomy education; it is central to how the discipline is taught. Students have historically gathered around a cadaver table, pointed at structures, discussed spatial relationships, and built clinical reasoning through peer interaction. When anatomy instruction moves to a digital surface, this collaborative dimension is often the first casualty: most digital tools are designed for individual use, and those that support shared displays do not account for the cognitive costs of content appearing upside down to some users or for the input conflicts that arise when multiple hands touch the screen simultaneously. Restoring the collaborative, around-the-table dynamic on a digital surface requires addressing orientation, input conflict, and gesture design together, not as isolated features but as an integrated interaction framework.

The most widely used 3D anatomy tools today include the Anatomage Table, Anatomy 3D Atlas, and Complete Anatomy, each with different pricing and accessibility models. Across these tools, three specific gaps remain unaddressed for collaborative tabletop learning. First, none offers a gesture vocabulary tuned to anatomical tasks; all rely on generic touchscreen gestures that do not capture domain-specific actions such as organ isolation or virtual dissection. Second, none provides content-orientation adaptation for users positioned around different sides of a shared display, a significant bar-

rier to legibility. Third, none includes software-level conflict resolution for simultaneous multi-user touch input, meaning that collaborative use is unsupported.

This study presents a multi-touch tabletop application designed for around-the-table anatomy learning on commodity touchscreen hardware. The work makes three contributions. First, a domain-specific gesture vocabulary was developed through a participatory design process involving medical students and anatomy instructors. Second, a content-alignment mechanism was implemented that automatically rotates displayed content based on the user's position around the table, with a manual override for locking orientation, addressing the legibility issue that arises when text and anatomical labels are viewed at high rotation angles. Third, a software-level conflict resolver was developed that enables two users to interact with the application simultaneously without requiring specialised hardware, closing the gap between hardware-bound multi-touch systems and commodity classroom displays. Together, these three components form a reusable, open-source framework that runs on standard multi-touch hardware, making collaborative 3D anatomy visualisation accessible to institutions where dedicated anatomy visualisation tables are not affordable.

The remainder of this study is organized as follows: the background and related work section examines the impact of 3D models in anatomy education and the requirements for collaborative gesture-driven tools, with a comparison of existing digital solutions, and identifies gaps that affect user experience. The methodology section presents the application's requirements, design, implementation, and the evaluation process used to assess its effectiveness. The results section presents the evaluation results of this work, followed by the discussion and conclusion.

II. Background and Related Work

A. The Role of 3D Models in Anatomy Education

Studies comparing 3D models with traditional methods for anatomy education generally support the combination of the two. Azer and Azer [8] report that incorporating 3D models into the teaching and learning process enhances the overall learning experience, increases productivity, and improves student satisfaction. Empirical comparisons point in the same direction. Yammine and Violato [7] reported that digital resources used alongside conventional techniques yield stronger learning outcomes than either approach alone.

A study by Brinke et al. [9] found that 46 undergraduate medical students at Erasmus University Medical Center who trained on 3D laparoscopic dissection models alongside lectures scored higher in both short-term and long-term anatomy assessments than peers receiving lecture-only instruction or dissection-based laparoscopic training. Murakami et al. [10] reported that 126 medical students and academic staff at the Graduate School of Medicine in Gunma University rated 3D anatomy-CT models positively, with significant improvement in radiology performance in subsequent clinical courses. A study by Müller-Stich et al. [11] compared the effectiveness of 2D CT images, static 3D-rendered liver models, and 3D models with rotational interaction (denoted 3Dr) for teaching and learning. 156 medical science students at the University of Heidelberg, Germany, were randomly assigned to one of three groups (2D, 3D, or 3Dr) and required to respond to a series of questions based on the lessons conducted. According to the results, the students in the 3D and 3Dr conditions performed significantly better than those in the 2D group. In 2024, Sarmiento [12] conducted a quantitative study to determine differences in written and lab exam scores between physical therapy students who received gross anatomy education using the Anatomage table and those who learned gross anatomy through traditional lectures. He reported a significant positive difference in written exam scores between Anatomage users and peers taught via traditional lectures.

Taken together, these studies indicate that gains from 3D visualization are largest when the comparator is textually or two-dimensionally constrained, extend beyond short-term recall to subsequent clinical reasoning, and persist when the 3D condition includes only static rendering.

B. The Need for Collaborative, Gesture-Driven Tools

Collaborative learning around a shared surface is central to how anatomy is taught, yet most digital tools are optimized for single-user interaction. Historically, supporting multiple users around a shared display required specialized hardware. Systems such as DiamondTouch [13] relied on antenna-embedded chairs to identify individual users via capacitive coupling. At the same time, other research has used overhead cameras to track hands using skin-color segmentation techniques [14]. These hardware-based solutions are often too expensive or complex for standard medical classrooms, particularly in resource-constrained environments where cost is a primary barrier to adoption [15]. A software-only approach to conflict resolution that works on commodity touchscreens would allow multiple students to interact simultaneously without specialized equipment.

The orientation of digital content creates an additional cognitive burden when students are positioned on opposite sides of the tabletop display. Human perception of text and objects is sensitive to rotation; for example, word recognition time is relatively stable for rotations up to 60°, but reaction times increase significantly as the orientation approaches 70–80% beyond 120°. Research indicates that at these higher angles, the brain shifts from holistic recognition to a much slower "letter-by-letter" processing method [16]. This "holistic rectification" threshold suggests that a visualization tool must automatically align its content to the user's position to remain legible and effective for learning.

Effective interaction also requires a gesture vocabulary that reflects the specific tasks of the medical domain. While standard mobile gestures like pinching and swiping are common, they often fail to capture complex anatomical actions such as organ isolation or virtual dissection. Research into the Surgery Planner system demonstrated that domain-specific interaction techniques improve usability and help maintain professional standards [17]. By using a participatory design approach, in which students and instructors help identify necessary gestures, developers can ensure alignment with users' mental models.

C. Current Digital Solutions and Limitations

Current anatomy software generally falls into two categories: high-end hardware-bound systems and mobile-centric

applications. Neither category fully addresses the needs of collaborative, tabletop learning.

The Anatomage Table is a touchscreen platform that serves as a virtual dissection table, using a library of 3D-reconstructed and prosections [18]. It allows for detailed interactive segmentation, enabling users to isolate specific muscular or neurovascular structures and view them across three section planes [18], [19]. While highly effective at simulating physical dissection, it does not provide software-based multi-user conflict resolution or adaptive content orientation. The Anatomage Table is a proprietary, non-portable hardware-software bundle with an acquisition cost that places it beyond the reach of most medical faculties in low and middle-income countries.

Anatomy 3D Atlas is an interactive virtual environment for learning about the body's various systems, including skeletal, circulatory, neurological, respiratory, digestive, and urinary systems. It supports basic 3D navigation and object manipulation, like hiding or multi-selecting organs [20]. However, it is primarily a single-user mobile tool that cannot display cross-sectional views or import medical scans such as CT or MRI data. The Anatomy 3D Atlas is subscription-based and designed for individual learning on mobile devices such as smarphones and tablets, not for shared-surface collaboration.

Complete Anatomy includes a highly detailed interactive atlas featuring over 13,000 structures and 12 systems [21]. It includes advanced features such as real-time muscle motion, nerve pathway tracing, and virtual dissection tools [22]. While it is a robust educational resource available on multiple operating systems, it is optimized for individual student use. It lacks collaborative features such as content alignment and multi-user touch handling on shared displays. When it comes to pricing, Complete Anatomy operates on a subscription model with tiered student, professional and institutional pricing [21].

Unlike educational tools, Surgery Planner was developed as a research prototype for clinical routine, specifically for planning orthopedic surgery [17]. It provides users with a comprehensive understanding of the surgical site, allowing them to draw and manipulate incision areas and explore various surgical options. While it proves the value of domain-specific gestures, it is a specialized planning tool rather than a general-purpose educational platform.

Table I highlights the specific research gaps identified among these tools. These differences mean that a direct feature comparison does not capture the full picture - features exclusive to a high-cost proprietary system may reflect its pricing tier rather than a technical advantage. Therefore this comparsion intends to identify functional requirements for collaborative tabletop anatomy learning that are unmet by any single existing tool, rather than to suggest that any individual tool is deficient within its own intended context.

The unmet functional requirements identified in this literature review informed the architecture of Anatomia. Most current applications rely on generic gestures and static content that do not account for the physical position of multiple users. There is a notable lack of a reusable, open-source framework that combines domain-specific gestures with content-alignment mechanisms. Anatomia fills this gap by providing a toolkit that addresses these collaborative needs while remaining accessible on commodity touchscreen hardware.

III. Methodology

This study followed a participatory design approach with summative task-based evaluation at the end of development. During the design phase, medical students and anatomy subject-matter experts iterated with the research team, and the application was shaped at each round to better meet learners' needs in dissection lessons. Four research questions guided the work:

- What are the suitable extended-hand gestures to interact with an anatomy visualization application?
- What are the suitable methods to address the orientation issues of the displayed content in order to enhance the user experience?
- What kind of mechanism is needed to handle the conflicts that occur when multiple users interact with the application simultaneously?
- How can a solution be provided as a framework that addresses the above-mentioned problems?

The methodology proceeded in four sequential phases: 1) problem identification and requirement gathering via the literature review and observational study of the target users; 2) solution design informed by domain expert input and a questionnaire distributed to medical students; 3) artifact development of a 3D anatomy visualization and interaction mechanism for large tabletop display systems; and 4) evaluation of the final product with medical and non-medical participants.

A. Domain Survey and Requirements

1) Observational user study: An observational study was conducted at the Faculty of Medicine, University of Wayamba, Sri Lanka, involving 75 students, 4 anatomy instructors, and 1 lecturer. At the time of the study, the faculty had only four cadavers for 150 students — approximately 1 cadaver per 38 students. Several observations were made during the study:

Most students could not clearly follow the dissection demonstrations. Only the two or three most confident students had the opportunity to perform the incision on the cadaver, leaving the rest of the group with little chance to observe the dissection. Even a small incision mistake on one cadaver could have a severe effect on all of the students allocated to that body. The same cadaver was used throughout the first academic year and was kept in glycerin rather than in freezers between classes. Smaller structures such as blood vessels, lymphatic vessels, and inner-ear organs were difficult to observe, and a few cadavers had fat layers or diseases that made certain organs hard to examine. Apart from a projector used to view some lessons with a camera, no infrastructure was available to assist in the lessons. Students reported consulting the Anatomy 3D Atlas on their personal mobile phones or laptops during dissection.

These observations established the gap that Anatomia was designed to fill: a tabletop visualization tool that supports co-located, multi-user study around a shared display without requiring additional cadaveric infrastructure.

TABLE I. COMPARISON MATRIX OF 3D ANATOMY APPLICATIONS

Feature	Anatomage	Anatomy 3D Atlas	Complete Anatomy	Surgery Planner	Anatomia
Virtual dissection	Available	Available	Available	Not available	Available
Domain-specific gestures	Not available	Not available	Not available	Available	Available
X-ray & CT image viewing	Available	Not available	Available	Available	Available
Multiple 3D models	Available	Available	Available	Not available	Not available
Details on parts	Available	Available	Available	Not available	Available
Multi-user interaction	Not available	Not available	Not available	Available	Available
Orientation change	Not available	Not available	Not available	Available	Available
Cross-platform	Not available	Available	Available	Not available	Available
Hardware platform	Proprietary tabletop display	Mobile devices	Personal computers & mobile devices	Multi-touch table	Commodity multi-touch display & mobile devices
Cost model	High (proprietary hardware –software bundle)	Subscription-based	Subscription-based	Research prototype	Open-source software; commodity hardware only

2) Gesture elicitation and requirements survey: A questionnaire was administered to 20 students from the Faculty of Medicine, University of Wayamba, and 3 medical science students from the Faculty of Medicine, University of Colombo. The questionnaire captured the suitable gestures for specific interactions, desired functional requirements, and features to avoid based on prior experience with existing solutions. Video clips and images were incorporated into the questionnaire to communicate the hand gestures under discussion.

For gesture identification, participants were asked to select the most suitable gestures for each of the following operations:

- Rotate the organ in the X, Y, and Z axes
- Drag the organ in the XY plane
- Zoom the 3D model in and out
- Select an organ or menu item
- Select multiple organs of the model
- Restore/rearrange the 3D model to its default orientation
- View the organ cross-section

Participants chose the appropriate gesture from nine options, including an ”Other” option that invited them to propose novel gestures, which the implementation team reviewed before adding them to the system vocabulary.

3) Domain expert input: According to the anatomy subject-matter expert consulted, anatomy lessons are generally conducted under separate sections of the human body, such as the head, neck, torso, upper extremities, and lower extremities, rather than by physiological system. To align with this pedagogical approach, the initial implementation of the application will focus on the human body by section rather than on systems as a whole, such as the digestive or skeletal systems.

4) Functional and non-functional requirements: Three main use cases were identified for the application: 1) select and view a 3D anatomical model, 2) view CT and X-ray images, and 3) interact with the 3D model through gestures. These use cases encompass sub-functionalities, including rotating and zooming the model, dragging it across the screen, viewing cross-sections, isolating and hiding organs, viewing organ details, and drawing annotations on the screen.

The application was designed to meet three key non-functional requirements: To ensure usability, it is tailored to a tabletop touch display. UI standards follow industry conventions, gestures map to real-world actions to aid memorization, with self-explanatory icons to facilitate faster learning. For accuracy, a conflict-resolution component was integrated to address conflicts arising from simultaneous multi-user interactions, and the application was designed to recognize and differentiate among hand gestures involving the same number of fingers. For performance, resource-efficient methods were adopted to implement resource-intensive functionalities, particularly when a 3D model is split into two parts.

B. System Architecture

Anatomia was built on a layered architecture that isolates hardware, operating system, rendering, and application logic as illustrated in Fig. 1.

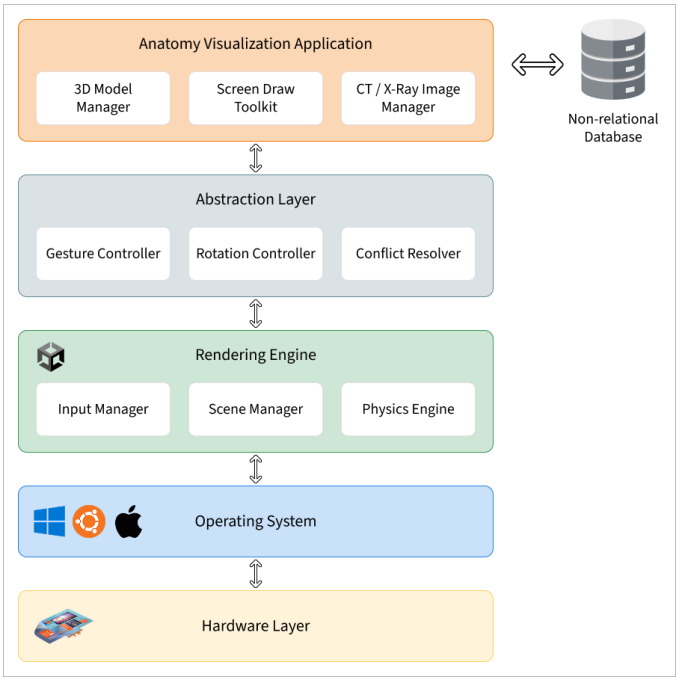


Fig. 1. System architecture diagram.

The hardware layer consists of the Displax Skin Ultra multi-touch display system, which can detect up to 100 simultaneous touch points and recognize objects. The operating system layer supports deployment on Windows, Ubuntu, and macOS. The rendering layer is powered by the Unity engine, which provides the input manager, scene manager, and physics engine. The abstraction layer contains reusable components that are loosely coupled to the anatomy visualization application: the gesture controller handles rotations, scaling, dragging, and cutting of 3D models; the rotation controller manages the rotation of UI elements to different orientations; and the conflict resolver addresses issues arising from simultaneous multi-user interaction. The application layer consists of three components: the 3D model manager, which displays user-selected 3D models and isolates relevant organs; the screen draw toolkit, which enables users to draw on the screen using single-handed gestures; and the CT/X-ray image manager, which facilitates the importing and viewing of CT and X-ray images. Organ information is stored in a MongoDB non-relational database.

C. Implementation

Anatomia was developed using Unity 2021.3 as the development environment, chosen for its rich asset store and built-in canvas system for UI development. Blender 3.1 was used to modify the 3D models due to its comprehensive feature set and free availability. The absence of fixed relationships or a specific set of attributes for organs made a non-relational database more suitable. Therefore, MongoDB 6.0 was selected as the database for storing information about human body systems and organs. The development language was C#, and version control was managed through GitHub. The application is compatible with Windows 7 and later, macOS, and Linux operating systems.

1) Gesture vocabulary: The survey outcomes and domain-expert input together produced a finalized gesture vocabulary. The survey outcomes and domain-expert input together produced a finalised gesture vocabulary. The design process applied three selection principles: 1) where a clear participant majority aligned with established touchscreen conventions, that gesture was adopted directly; 2) where participants selected the same gesture for multiple operations, an alternative gesture was designed in consultation with participants and domain experts to resolve the conflict; and 3) where participants favoured the "Other" option over the predefined gestures, the most intuitive gesture from the participant-proposed suggestions was selected in consultation with domain experts.

Applying the first principle, rotation along the X, Y, and Z axes was assigned to single-finger drag, which carried the highest preference share in each case (Fig. 2a, Fig. 2b, Fig. 2c) and matches the conventional touchscreen drag-to-rotate mapping. For dragging the 3D model across the XY plane, the five-finger drag captured the majority of preferences at 39.1% (Fig. 2d). Zoom was assigned to two-finger pinch, chosen by 43.5% of participants (Fig. 2e), consistent with the established pinch-to-zoom convention. As shown in Fig. 2f, restoring the 3d model to the default anterior view was assigned to a single-finger double tap, which held the majority preference at 34.8%. These gestures required no adaptation beyond implementing the conventional mapping.

Single-finger tap was assigned to single organ or menu item selection, preferred by 82.6% of participants (Fig. 2g). The same gesture also received greater preference for selecting multiple organs (Fig. 2h). To resolve this conflict, the second principle was applied, and a two-stage gesture was designed: placing a single finger on an organ activates multi-select mode, after which subsequent single-finger taps add organs to the selection.

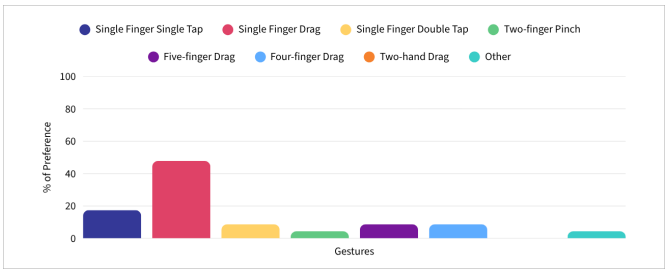
For viewing the cross-section of the model, the largest share of participants selected the "Other" option (Fig. 2i), indicating that none of the standard gestures presented in the questionnaire felt appropriate for this anatomically specific action. Participants who selected "Other" were invited to propose alternative gestures. Among the proposed suggestions, domain experts were consulted to identify the most intuitive option. The selected gesture involved the user placing all five fingers steadily on the screen while dragging a single finger across the 3D model in the area where the cross-section is to be observed. This gesture was deemed appropriate based on the input provided by the domain experts, as it mirrors the physical action of holding a specimen steady with one hand while cutting with the other, a mapping considered intuitive for students with dissection experience.

The gesture vocabulary described above represents the final iteration of the design, which was refined through formative evaluation during development. In an initial round of user testing, participants expressed difficulty remembering the specific gestures required for certain actions. They also suggested that a visual guide or tutorial on the gesture mappings could be helpful, particularly for users who are new to the system or who may not be familiar with the gestures used in the application. In response to this feedback, the gesture mappings were revised to ensure clearer differentiation between gestures sharing the same initial hand posture. A visual guide displaying the gesture vocabulary was also integrated into the application interface, providing a reference that users could consult during interaction.

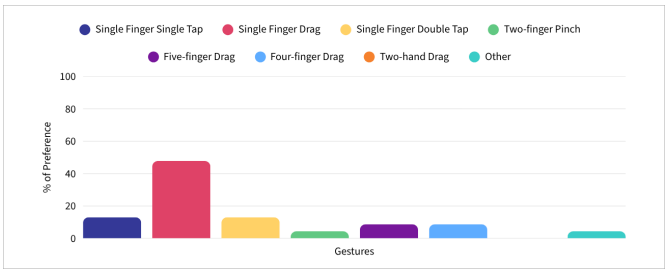
The gesture controller was implemented to handle all hand gestures for the application. It comprises a primary script that orchestrates gesture recognition, a touch controller that manages single and double taps, helper scripts for auxiliary functions, and customizable scripts for each gesture that facilitate 3D object and UI manipulation. These customizable scripts can be added to any 3D object and configured with corresponding parameters to execute a specific gesture, enabling the definition of additional gestures.

2) Content alignment mechanism: A radial menu at one corner of the screen indicates the user's position relative to the display. The mechanism provides auto-rotation from the menu position and a manual override that locks the orientation. Boundary parameters for auto-rotation can be reconfigured in the Unity editor without manual code changes. The threshold was set at 60°, following Koriat & Norman's finding that reading rates stay stable up to that angle [16].

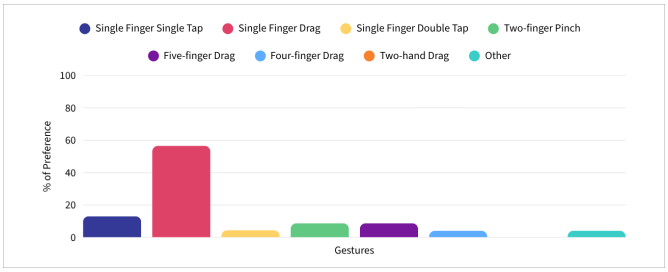
To ensure optimal content orientation, it is necessary to determine the user's position relative to the display. The proposed content alignment mechanism comprises two primary features. The first is automatic content rotation based on the user's position, achieved by leveraging the application's radial



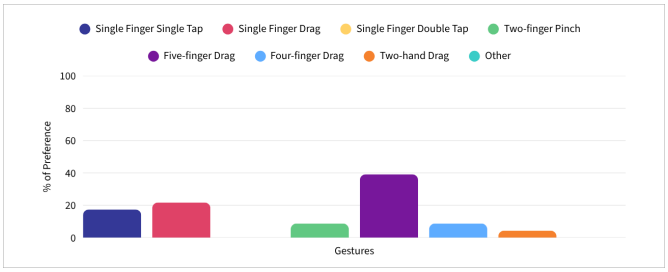
(a) Gesture preference for rotating an organ around the X-axis.



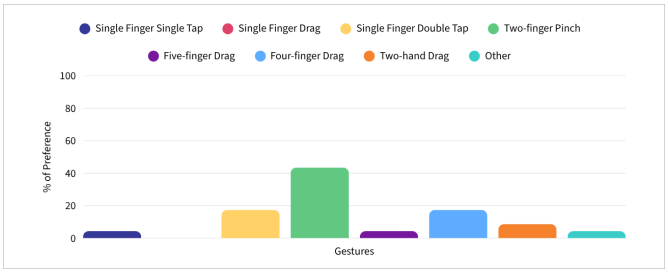
(b) Gesture preference for rotating an organ around the Y-axis.



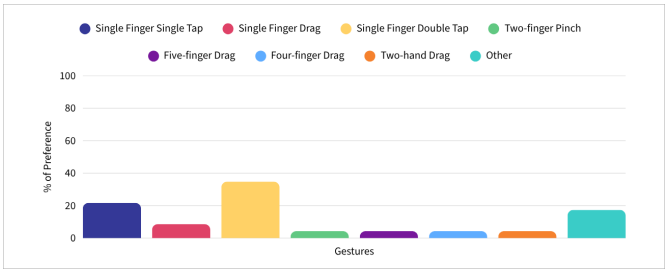
(c) Gesture preference for rotating an organ around the Z-axis.



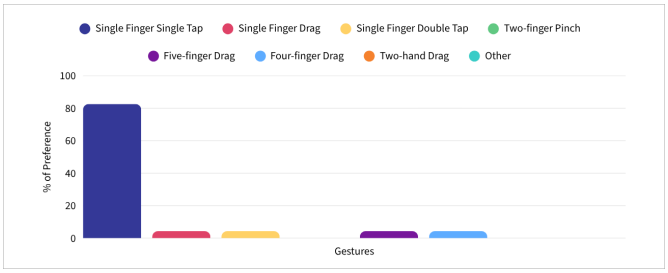
(d) Gesture preference for dragging an organ in the XY plane.



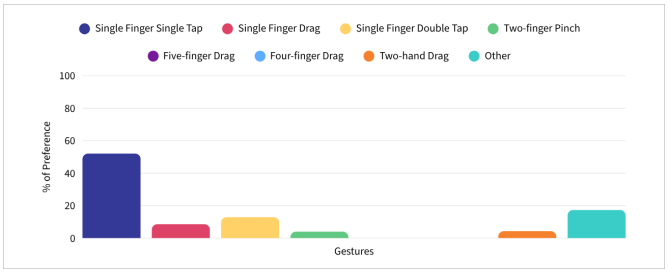
(e) Gesture preference for zooming the 3D model in/out.



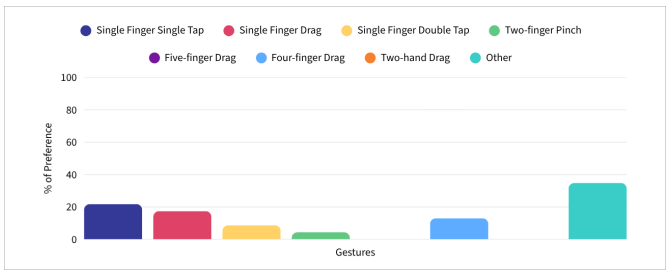
(f) Gesture preference for restoring the 3D model.



(g) Gesture preference for selecting an organ/menu item.



(h) Gesture preference for selecting multiple organs of the model.



(i) Gesture preference for viewing organ cross-section.

Fig. 2. Gesture preferences for 3D model interactions.

menu. The approach assumes that the user positions the radial menu on their side of the screen; the content on the display is then oriented accordingly. The radial menu can be moved across the screen, and all UI elements rotate accordingly. The auto-rotation boundary parameters can be easily modified in the Unity editor without requiring manual code changes. The second feature allows users to manually rotate the UI components in the X and Y planes and lock them in place to prevent auto-rotation, providing flexibility for specific viewing preferences.

3) Multi-user conflict resolution: Identifying users' touches distinctly on a multi-touch surface has traditionally required specialized hardware infrastructure, such as DiamondTouch's antenna-embedded chairs [17] or camera-based hand-tracking systems [14]. The focus of this project was to address multi-user interaction issues by resolving conflicts of the hand gestures of two simultaneous users at the application level, without requiring any additional hardware.

A conflict-resolution component was introduced that operates alongside an accessibility button. The conflict resolver operates as follows for two simultaneous users: 1) the screen is divided into two equal parts; 2) once a user touches the screen, the conflict resolver checks the number of touches on that particular side of the screen; 3) based on the number of touches and the behavior of those touches, the corresponding gesture is performed; and 4) if both users attempt to manipulate the 3D model at the same time, priority is given to the action performed by the user who has the radial menu or accessibility button on their side. This method is entirely implemented at the software level, with no additional hardware components required. UI dragging, 3D model rotation, and 3D model zoom-in/out actions were implemented and tested using this conflict-resolution mechanism. Other gestures can also be extended by following the same approach to provide a multi-user experience.

4) Object dissection: The 3D anatomy model used in the application was a combination of mesh models, and two approaches for implementing virtual dissection were investigated: mesh cutting and clipping.

The mesh cutting technique is performed with a cut gesture, in which one hand holds the model while the other draws a line. A line renderer responds to finger movement. The algorithm for separating the 3D model proceeds as follows: two empty game objects are created to represent the two sides of a given cutting plane. For each mesh model, the algorithm checks whether it intersects with the cutting plane. If it does not intersect, the model is added as a child game object to the corresponding side. If it does intersect, two dynamic mesh models are constructed (one for each side of the plane), and the position of each polygon relative to the cutting plane is computed to assign it to the appropriate side.

The clipping method, by contrast, does not divide the 3D mesh model into two separate parts. Instead, it uses a clipping plane, a custom shader, and materials created with that shader to hide or make transparent a portion of the 3D mesh. A line is drawn using the line renderer in response to finger movement; the clipping plane changes its position to the line location and orients itself to the same angle; and the user selects the part that must be removed or hidden, which becomes transparent (Fig. 3). Because clipping requires fewer computational resources compared to mesh cutting, it was chosen as the feasible approach for the application.

5) X-Ray/CT image viewer: Anatomia enables users to access and interact with up to two X-ray or CT images simultaneously. Users can manipulate the images through three gestures: two-finger rotation, which allows the user to hold one finger in a fixed position on the image while moving another finger to rotate the image in any direction; two-finger zoom, which enables the user to zoom in or out by moving two fingers in opposite directions; and five-finger drag, which allows dragging the images across the screen in any direction.

6) Other 3D object manipulation features: Several additional features were implemented to enhance the learning experience of Anatomia.

The drawing tool lets users draw on the screen with a one-finger gesture to annotate or mark specific areas in images or 3D models. A line can be drawn by dragging one finger across the screen, and users can select from a predefined set of colors to draw around the 3D model, X-ray, and CT images.

The organ detail viewer allows users to select an organ and view a pop-up with basic information about it (Fig. 4). The hide and isolate functionality allows users to select one or more organs with a single tap or a long-tap gesture, then either isolate the selected organs (viewing only those) or hide specific organs from view. This functionality employs four stacks to manage the display state: an undo stack that stores all removed organs at each step, a redo stack that stores organs added during each undo step, an undo sequence stack that tracks the number of organs removed at each action, and a redo sequence stack that tracks the number of organs restored at each action.

The layer-wise view feature provides navigation through the multiple layers of the 3D model, with each organ placed precisely under a single layer. The Layers UI allows users to hide or restore previously hidden layers using the "Add" and "Hide" (5) options, and the "Select" option enables or disables the desired layers. The application maintains a list of layers, with two Boolean variables for each: one indicating whether the layer is entirely hidden when the "Select" option is used, and another indicating whether the layer is temporarily hidden when the "Add" or "Hide" option is used (Fig. 5).

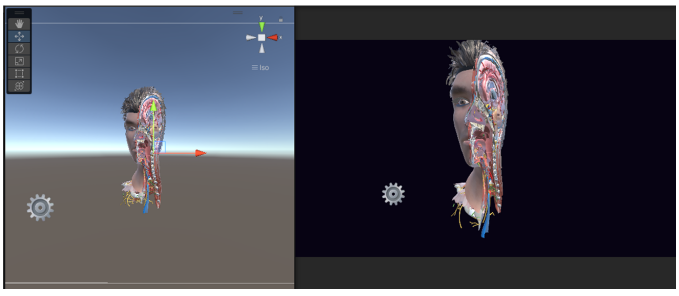


Fig. 3. Object dissection via clipping.

D. Evaluation

The application underwent both formative and summative evaluation. Formative evaluation was conducted iteratively throughout the development cycle to refine the interface based on user feedback. Summative evaluation assessed the user experience of the developed 3D anatomy visualization application. Two complementary analyses were carried out: a performance metrics analysis of task success and a self-reported metrics analysis comparing Anatomia with the Anatomy 3D Atlas and traditional cadaver dissection.

For the performance metrics analysis, two groups of participants were recruited: 1) 8 students from the UOC Faculty of Medicine and the Faculty of Indigenous Medicine who had prior experience with traditional cadaveric dissection, used to evaluate the application's fit for experienced anatomy learners; and 2) 5 students from non-medical backgrounds, used to evaluate usability, learnability, and collaboration support for novice users. Performance was measured using a task-success metric: the proportion of tasks each participant completed successfully out of 15 administered tasks covering hand gestures, content alignment, and multi-user interaction capabilities (6). The task list is shown in Table II; their IDs are referenced throughout the remainder of the study.

This analysis was conducted in two phases to record task performance both with and without assistance. Phase 1 established the unassisted baseline for each participant, and Phase 2 established performance after brief verbal guidance. In Phase 1, each participant performed every task in Table II in sequence, without assistance from the evaluator. A think-aloud protocol was used throughout this phase to record expectations, misconceptions, and gesture-association patterns alongside task-success scores. In Phase 2, the evaluator gave brief oral instructions to any participant who had failed a task in Phase 1, and the participant then repeated the task. This phase established the level each participant could reach with minimal verbal guidance rather than independent exploration. By comparing Phase 1 and Phase 2 performance for each task, the evaluation distinguishes initial learnability from overall usability. Task success was scored on a five-level scheme as follows, with one score recorded per participant per task:

- **Score 1.0**: Participant completed the task on the first attempt without any assistance.

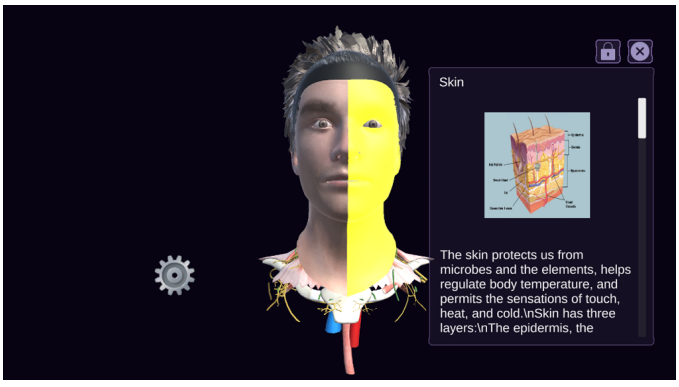


Fig. 4. View organ details.

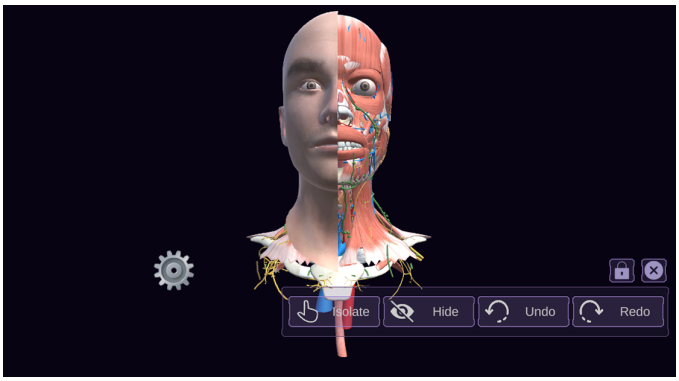


Fig. 5. View after hiding an organ/layer.

- **Score 0.75**: Participant failed the first attempt but succeeded on the second with minor verbal guidance.
- **Score 0.5**: Participant failed the first two attempts but succeeded after observing a group member perform the task.
- **Score 0.25**: Participant failed several attempts and eventually succeeded after observing a group member perform the task.
- **Score 0**: Participant failed to complete the task after more than three attempts.

Descriptive statistics were used to analyze the performance metrics ratings, including mean task-success scores across participants and the frequency distribution of each score level per task [23]. Given the ordinal nature of the task-success scores and the small sample sizes, non-parametric tests were used for all inferential analyses (Fig. 6). Between-group comparisons (medical vs. non-medical) were conducted using the Mann–Whitney U test [24], with Cliff's delta (δ) as the effect-size measure [25]. Cliff's delta quantifies the degree to which scores in one group tend to exceed scores in the other, ranging from -1 to $+1$; absolute values below 0.147 are interpreted as negligible, between 0.147–0.33 as small, between 0.33–0.474 as medium, and at or above 0.474 as large [26]. Within-group comparisons of familiar versus novel gestures were analysed using the Wilcoxon signed-rank test, with effect size r computed as the standardised test statistic divided by the square root of the number of pairs. All tests were two-tailed at $\alpha = 0.05$.

The Self-reported metrics analysis was carried out to identify users' perception of the application and their interactions [27]. This included a post-session questionnaire with 5-point Likert-scale statements (strongly disagree, disagree, neutral, agree, strongly agree) and one open-ended prompt for free-text suggestions. This questionnaire was designed specifically for this evaluation to capture comparative judgments between Anatomia, the Anatomy 3D Atlas, and traditional cadaver dissection. The questions were informed by the usability dimensions identified during the participatory design phase but should be interpreted as exploratory indicators of user perception rather than psychometrically validated measures. Each participant was allowed to interact with both applications. Then they were provided with a questionnaire that tested

TABLE II. TASKS EVALUATED IN THE PERFORMANCE METRIC ANALYSIS

Task ID	Task	Scenario
T1	Inspect an organ: zoom in or out of a particular organ	Hand gestures
T2	Rotate an organ around the X, Y, and Z axes	Hand gestures
T3	Select a single organ to view its details	Hand gestures
T4	Select multiple organs to isolate or hide them as a group	Hand gestures
T5	Restore the 3D model to its default anterior view after rotation	Hand gestures
T6	Drag the 3D model across the XY plane	Hand gestures
T7	Rotate a UI element (pop-up) manually	Hand gestures
T8	Drag a UI element across the screen	Hand gestures
T9	Upload an X-ray image to the X-ray viewer and perform zoom, drag, and rotate	Hand gestures
T10	Dissect the 3D model and view a cross-section	Hand gestures
T11	Rotate all visible UI elements to the opposite direction	Content alignment
T12	Disable auto-rotation of a particular UI element	Content alignment
T13	Perform two zoom actions simultaneously from opposite sides of the screen	Multi-user
T14	Perform zoom on the 3D model and rotate the model simultaneously	Multi-user
T15	Perform zoom on the 3D model and drag a UI element simultaneously	Multi-user

whether the Anatomia application adds pedagogical value and whether it is more usable than alternative tools, such as the Anatomy 3D Atlas application and cadaver dissection. The phenomenon in which participants tend to provide responses that they believe will be perceived favorably by others is commonly known as social desirability bias [28]. To mitigate this bias, the questionnaire was anonymous, and all 13 participants completed it without being observed.

Before the summative evaluation, the Anatomia application was tested at four levels: unit testing of gesture functions using Unity Remote 5; integrated testing across the mobile device and the tabletop display system; system testing through preliminary user studies; and performance testing to ensure resource efficiency under concurrent multi-user interaction.

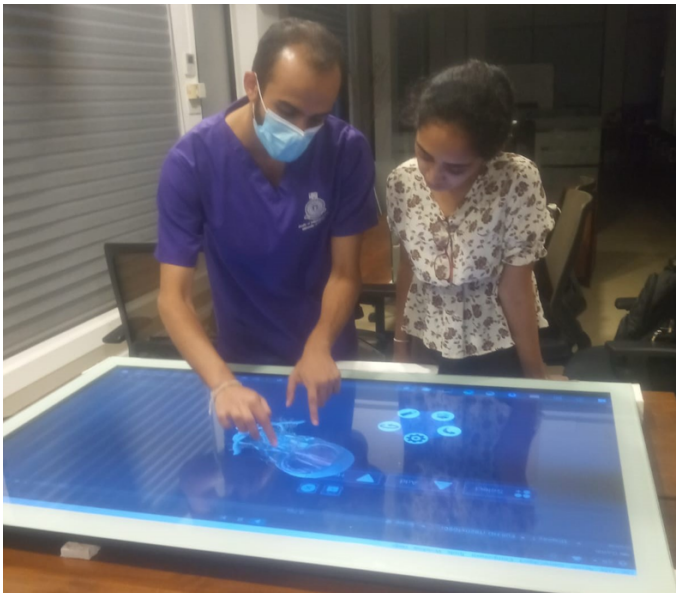


Fig. 6. Medical science students performing tasks in the summative evaluation.

IV. RESULTS

A. Performance Metrics Analysis Results

Medical students (Tables III and V): The 8 medical students achieved an overall median task-success score of 0.90 (M = 0.89, SD = 0.04) across all 15 tasks. Tasks drawing on conventional touchscreen gestures such as single-tap selection (T3, Mdn = 1.00), single-finger drag for rotation (T2, Mdn = 1.00), and two-finger pinch to zoom (T1, Mdn = 1.00) received near-perfect scores, as did T8 and T9, which involve familiar gestures applied to UI elements and image upload. A Wilcoxon signed-rank test confirmed that familiar touchscreen gestures (Mdn = 1.00) yielded significantly higher scores than the novel domain-specific gestures (Mdn = 0.75) within this group (W = 0, p = .008, r = 0.89, large effect). The distribution of score levels across the medical-student group is shown in Fig. 7.

The lower scores corresponded to the anatomically novel gestures: five-finger drag for model translation (T6, Mdn = 0.75), single-finger double tap for default-view restoration (T5, Mdn = 0.75), and the five-finger steady with one-finger drag for cross-section viewing (T10, Mdn = 0.63). For T5 and T6, most participants failed on the first attempt but succeeded on the second after minimal verbal guidance. The implications of this pattern for learnability versus mental-model alignment are examined in Section V. Content-alignment tasks (T11, Mdn = 0.75; T12, Mdn = 1.00) and multi-user interaction tasks (T13–T15, all Mdn = 1.00) recorded scores at or near ceiling for the medical-student group, with zero variance across participants on the multi-user tasks, indicating that the auto-rotation mechanism and the conflict resolver added no measurable usability overhead.

Non-medical students (Table IV and V): The 5 non-medical students recorded an overall median task-success score of 0.83 (M = 0.82, SD = 0.04), lower than the medical group. A Mann–Whitney U test on participant-level averages confirmed that this overall group difference was statistically significant (U = 37.0, p = .015, Cliff's δ = 0.85, large effect), consistent with a carryover advantage from prior specimen-handling experience among medical students.

A detailed breakdown of participant-level scores is shown in Fig. 8. The largest differences from the medical group

TABLE III. TASK SUCCESS METRICS OF MEDICAL STUDENTS

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	Mean
P1	1.0	1.0	1.0	0.75	0.75	0.75	0.75	1.0	1.0	0.5	0.75	1.0	1.0	1.0	1.0	**0.88**
P2	1.0	1.0	1.0	0.75	0.75	1.0	0.75	1.0	1.0	0.5	0.75	1.0	1.0	1.0	1.0	**0.9**
P3	1.0	1.0	1.0	0.75	1.0	1.0	0.75	1.0	1.0	0.75	0.75	1.0	1.0	1.0	1.0	**0.93**
P4	0.75	1.0	1.0	0.5	0.75	0.5	0.5	1.0	1.0	0.5	0.75	1.0	1.0	1.0	1.0	**0.82**
P5	1.0	1.0	1.0	0.75	0.75	0.75	0.75	1.0	1.0	0.75	0.75	1.0	1.0	1.0	1.0	**0.9**
P6	1.0	1.0	1.0	1.0	0.75	0.75	0.75	1.0	1.0	0.75	0.75	1.0	1.0	1.0	1.0	**0.92**
P7	1.0	1.0	1.0	0.75	0.75	0.75	0.75	1.0	1.0	0.5	0.75	1.0	1.0	1.0	1.0	**0.88**
P8	1.0	1.0	1.0	1.0	0.75	0.75	0.75	1.0	1.0	0.75	0.75	1.0	1.0	1.0	1.0	**0.92**
Mean	**0.97**	**1.0**	**1.0**	**0.78**	**0.78**	**0.78**	**0.78**	**1.0**	**1.0**	**0.63**	**0.75**	**1.0**	**1.0**	**1.0**	**1.0**	**0.89**

TABLE IV. TASK SUCCESS METRICS OF NON-MEDICAL STUDENTS

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	Mean
P1	1.0	1.0	1.0	0.75	0.75	0.75	0.75	1.0	0.75	0.5	0.75	1.0	1.0	1.0	1.0	**0.87**
P2	0.75	1.0	0.75	0.5	0.75	0.5	0.5	0.75	0.75	0.5	0.75	1.0	1.0	1.0	1.0	**0.77**
P3	1.0	0.75	1.0	0.75	0.75	0.75	0.5	1.0	0.75	0.5	0.75	1.0	1.0	1.0	1.0	**0.83**
P4	0.75	0.75	0.75	0.5	0.75	0.75	0.75	1.0	0.75	0.5	0.75	1.0	1.0	1.0	1.0	**0.8**
P5	1.0	1.0	1.0	0.5	0.75	0.75	0.75	1.0	0.75	0.5	0.75	1.0	1.0	1.0	1.0	**0.85**
Mean	**0.9**	**0.9**	**0.9**	**0.6**	**0.75**	**0.7**	**0.65**	**0.95**	**0.75**	**0.5**	**0.75**	**1.0**	**1.0**	**1.0**	**1.0**	**0.82**

TABLE V. MANN-WHITNEY U TEST RESULTS: MEDICAL VS. NON-MEDICAL STUDENTS

Task ID	Scenario	Med Mdn	Non-Med Mdn	U	p	Cliff's δ	Effect
T1	Familiar hand gesture	1.00	1.00	25.5	0.317	0.275	Small
T2	Familiar hand gesture	1.00	1.00	28.0	0.080	0.400	Medium
T3	Familiar hand gesture	1.00	1.00	28.0	0.080	0.400	Medium
T4	Novel hand gesture	0.75	0.50	31.5	0.075	0.575	Large
T5	Novel hand gesture	0.75	0.75	22.5	0.527	0.125	Negligible
T6	Novel hand gestures	0.75	0.75	25.5	0.369	0.275	Small
T7	Novel hand gesture	0.75	0.75	25.5	0.317	0.275	Small
T8	Familiar hand gesture	1.00	1.00	24.0	0.268	0.200	Small
T9	Familiar hand gesture	1.00	0.75	40.0	< 0.001	1.000	No variance
T10	Novel hand gesture	0.62	0.50	30.0	0.083	0.500	Large
T11	Content alignment	0.75	0.75	20.0	1.00	0.000	No variance
T12	Content alignment	1.00	1.00	20.0	1.00	0.000	No variance
T13	Multi-user interaction	1.00	1.00	20.0	1.00	0.000	No variance
T14	Multi-user interaction	1.00	1.00	20.0	1.00	0.000	No variance
T15	Multi-user interaction	1.00	1.00	20.0	1.00	0.000	No variance
Overall	**All**	**0.90**	**0.82**	**37.0**	**.015**	**0.85**	

appeared on the anatomically demanding tasks: T4 (multi-organ select, Mdn = 0.50, δ = 0.58) and T10 (cross-section viewing, Mdn = 0.50, δ = 0.50), both showing large effect sizes. However, neither per-task comparison reached significance at α = 0.05 (T4: U = 31.5, p = .075; T10: U = 30.0, p = .083), which is consistent with the limited statistical power of the sample rather than evidence of no group difference. T5 through T7 ranged from Mdn = 0.75 to Mdn = 0.65. With minimal verbal guidance during Phase 2, the non-medical participants completed most failed tasks on the next attempt. T10 recorded the lowest score in the non-medical group, with all five participants requiring peer observation before completing the task. The possible explanations for this pattern are discoverability, mental-model mismatch, or domain unfamiliarity. The pattern across both groups indicates that participants achieved task completion across the full gesture set within a single guided session, with the largest performance gaps concentrated on the anatomically novel gestures rather than the interface mechanisms. Content-alignment and multi-user tasks showed identical median scores across both groups (T11: Mdn = 0.75; T12–T15: Mdn = 1.00), confirming that these mechanisms performed consistently regardless of domain background.

B. Self-Reported Metrics Analysis Results

The post-session questionnaire aimed to capture participants' comparative judgments of Anatomia relative to the Anatomy 3D application and traditional lessons. Fig. 9–10 present the bar-chart distributions for the six reported items, and the remaining items are summarised at the end of this subsection. All response frequencies are reported as a numerator over 13 participants (N=13).

In the questionnaire-based comparison, participants consistently rated Anatomia favorably over both traditional dissection lessons and the Anatomy 3D Atlas. On the user-friendliness

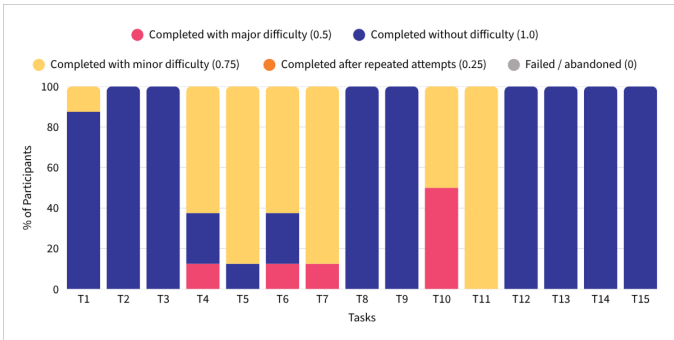


Fig. 7. Task success rates of medical students.

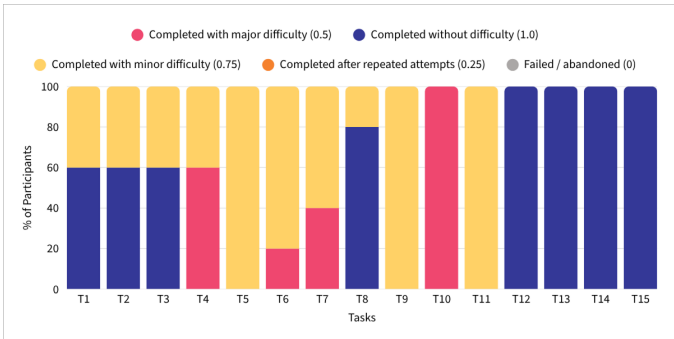


Fig. 8. Task success rates of non-medical students.

of Anatomia over Anatomy 3D Atlas, all 13 participants (8 strongly agree, 5 agree; 100.0% total agreement) endorsed Anatomia, as shown in Fig. 9a. On around-the-table alignment, again 13 out of 13 (6 strongly agree, 7 agree; 100.0% agreement) voted for Anatomia (Fig. 9b). Regarding menu design, all participants (9 strongly agree, 4 agree; 100.0% agreement) endorsed the radial menu over the Atlas fixed menus (Fig. 9c). Anatomia's ability to allow multi-user interaction was also endorsed unanimously (8 strongly agree, 5 agree; 100.0% agreement, as shown in Fig. 9d). This complements the performance-metric findings in Section IV-A, where tasks involving multi-user interaction reached 1.00 across all 13 participants.

When comparing the Anatomia application to cadaver-based dissection, regarding reusability, all 13 participants (9 strongly agree, 4 agree; 100.0% agreement) endorsed Anatomia as superior to traditional dissection (Fig. 10a). On collaboration, 10 of 13 (5 strongly agree, 5 agree; 76.9% agreement) endorsed Anatomia, with 2 of 13 remaining neutral and 1 of 13 strongly disagreeing (Fig. 10b). The gap between reusability and collaboration aligns with the cadaver-access constraints recorded at the University of Wayamba during the observational study (150 medical students shared 4 cadavers across the academic year): reusability addresses a constraint that participants experience universally, whereas the collaboration axis compares Anatomia's deliberate two-user protocol against the unstructured peer interaction that already happens at a single cadaver station.

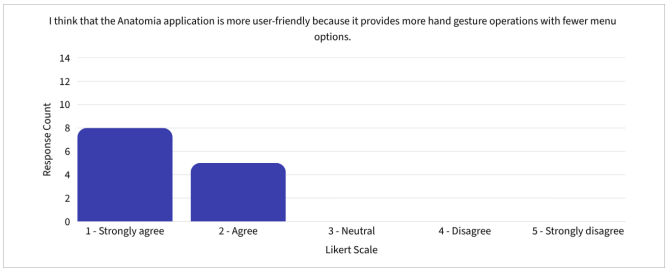
V. Discussion

The summative evaluation supports each of the three design contributions outlined in Section III: the gesture vocabulary is learnable, the content alignment mechanism scores highly on usability, and the software-level conflict resolver works in the context of two concurrent user interactions without introducing qualitative friction. The overall group difference between medical students (Mdn = 0.90) and non-medical students (Mdn = 0.82) was statistically significant (U = 37.0, p = .015, Cliff's δ = 0.85), and the self-reported comparison items reached 100.0% or 76.9% agreement, providing evidence behind that conclusion.

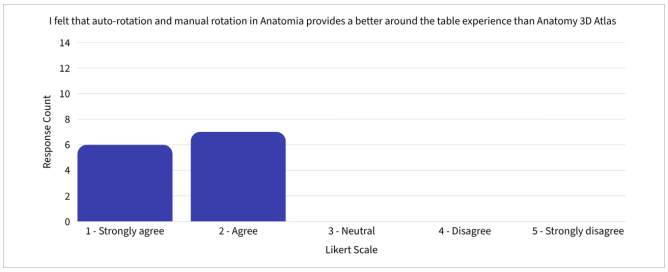
The three basic hand-gesture tasks—single-finger tap for selection, single-finger drag for rotation, and two-finger pinch for zoom—recorded first-attempt task-success averages of 0.97, 1.00, and 1.00 in the medical-student group (Table III) and equivalent values in the non-medical group (Table IV). The anatomically novel gestures recorded lower first-attempt scores: T4 (multi-organ select) averaged 0.6 in the non-medical group; T5 (default-view recall), T6 (five-finger drag), and T7 (manual UI rotation) ranged from 0.65 to 0.75; and T10 (cross-section viewing) recorded the lowest single-task score, 0.5, in the non-medical group, meaning that participants succeeded only after observing a peer perform the gesture. Because the composite scoring scale does not separate discoverability from mental-model alignment, this result is open to two interpretations: the gesture may be learnable but not independently discoverable from the interface cues alone, or participants' prior expectations of how cross-sectional viewing should work may have been incompatible with the five-finger steady plus one-finger drag design. The fact that medical students also scored below ceiling on T10 (Mdn = 0.63) suggests that discoverability is the primary barrier, though the current instrument cannot confirm this distinction. For the remaining gesture tasks, content-alignment tasks, and multi-user tasks, effect sizes were small or negligible, indicating that the group difference does not reflect a structural usability gap. Both groups converged on task completion after a single round of verbal guidance.

These findings establish that the gesture vocabulary is learnable within a single guided session, but they do not speak to long-term retention. Learnability and retention are distinct constructs: a gesture that is easy to acquire during an introduction may still be difficult to recall after a period of disuse, particularly if it lacks a strong mnemonic mapping to a familiar physical action. The novel gestures in Anatomia, such as the five-finger steady plus one-finger drag for cross-sectional viewing, are designed to mirror anatomical actions, which may support longer-term recall, but this remains an empirical question. A follow-up evaluation conducted after two to four weeks without practice would be necessary to determine whether the proposed interaction techniques are retained or whether periodic reinforcement is needed.

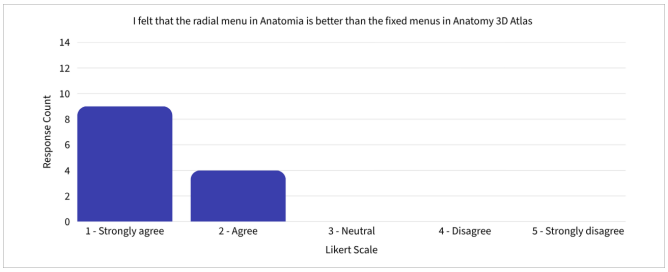
Koriat and Norman showed that reading time remains stable through rotations up to 60° and degrades substantially beyond 120° of in-plane rotation, where the visual system shifts from holistic to letter-by-letter processing [16]. Existing commercial systems reviewed in I did not address this dimension. The dual approach adopted in Anatomia pairs automatic rotation from the radial-menu position with manual rotation



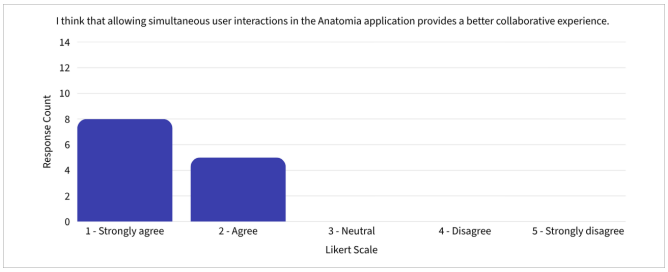
(a) Anatomy 3D Atlas vs. Anatomia application's hand gestures.



(b) Anatomy 3D Atlas vs. Anatomia in terms of content alignment.

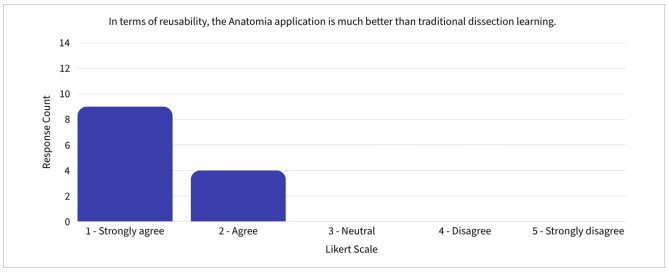


(c) Anatomy 3D Atlas vs. Anatomia in terms of fixed menu options.

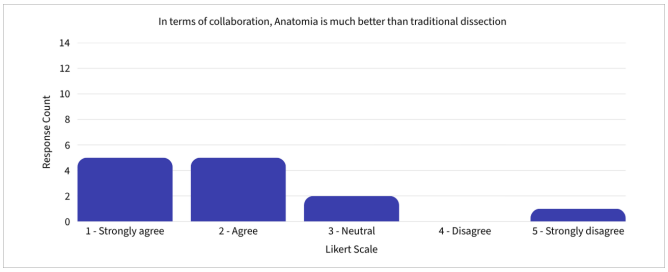


(d) Anatomy 3D Atlas vs. Anatomia in terms of multi-user interaction.

Fig. 9. Comparison of anatomia application against anatomy 3D atlas.



(a) Traditional lessons vs. Anatomia application in terms of reusability.



(b) Traditional lessons vs. Anatomia application in terms of collaboration.

Fig. 10. Comparison of anatomia application against traditional dissection lessons.

and locking. The alignment task set (T11–T12) recorded a median score of 0.88 in both groups, with zero between-group variance (Cliff's δ = 0.00), indicating that the mechanism adds little cognitive overhead while removing the legibility issue of fixed orientation. The identical performance across groups suggests that content alignment is intuitive irrespective of anatomical background.

The self-reported metric questionnaire reinforces this finding: both statements-"auto/manual rotation provides a better around-the-table experience than the fixed orientation of the Anatomy 3D Atlas," and "the radial menu is better than fixed menus"-received 100% agreement, with zero neutral or disagreeing responses. This is consistent with the underlying reading-time penalty: participants who have experienced the legibility drop on a fixed-orientation app perceive the rotation mechanism as an unambiguous improvement.

Prior multi-user identification mechanisms on shared surfaces require hardware [17], [14] that is not widely available in teaching classrooms. The University of Wayamba's anatomical resources had 4 cadavers for 150 medical students during the academic year of the observational study, resulting in an approximately 38:1 student-to-cadaver ratio, which makes capital investment in specialized hardware a binding adoption constraint. The conflict resolver in Anatomia runs at the software level. Multi-user task success reached a perfect 1.00 (Mdn = 1.00, zero variance) across both participant groups, with no between-group difference (Cliff's δ = 0.00). The corresponding self-reported item received 13 of 13 endorsements (100.0% agreement). The convergence of ceiling-level task success and unanimous subjective endorsement across participants with different domain backgrounds provides consistent evidence that the conflict-resolution mechanism does not impose a usability penalty.

This study has several limitations. The participant sample comprised 13 individuals (8 medical, 5 non-medical) drawn from two faculties of a Sri Lankan university, which constrains the generalisability of the results to other populations, educational systems, and cultural contexts. Given these sample sizes, the non-parametric tests employed have sufficient power

to detect only large effects. The per-task Mann–Whitney U comparisons did not reach significance at $\alpha = 0.05$, even where effect-size estimates indicated large group differences (e.g., T4: Cliff's $\delta = 0.58$; T10: $\delta = 0.50$). These non-significant per-task results should be interpreted as inconclusive rather than as evidence of no difference between groups. Replication with a larger and more demographically diverse cohort would be necessary to detect small and medium effects and to establish external validity. Recent cross-cultural studies have raised the standard required for the methodology used in usability investigations of interactive educational applications. Criollo-C et al. [29] assessed a mobile augmented reality application designed to provide training on PC hardware in three countries using the IBM Computer System Usability Questionnaire (CSUQ), a well-established tool that has proven psychometric properties. Their study highlights the value of employing validated measurement instruments and of using samples that are geographically diverse when evaluating educational technology.

The summative evaluation benchmarked Anatomia against the Anatomy 3D Atlas and traditional cadaver dissection, but not against the Anatomage Table, which Table I identifies as the closest existing tabletop competitor. The Anatomage Table's acquisition cost made a controlled academic comparison impractical within the study's resource constraints. A controlled comparison against the Anatomage Table would provide a more balanced evaluation of Anatomia's contributions.

Anatomia's current implementation includes a single 3D model covering only the head region, whereas all three commercial competitors support multiple 3D models spanning the full body, a capability that Anatomia does not yet offer. This limits the evaluation's ecological validity: gesture challenges specific to larger or more complex anatomical regions are not represented in the current task set. The system architecture was designed to accommodate additional models through the 3D Model Manager component, but the feasibility of maintaining acceptable frame rates and interaction responsiveness as model complexity and polygon counts increase has not been empirically verified. Future work should assess how the framework scales to more detailed 3D models, larger anatomical datasets, and varying hardware configurations.

Finally, this study measured interaction quality, usability, and learnability, not learning outcomes. Whether the interaction-level gains documented here translate into measurable improvements in anatomical knowledge over a curriculum term remains open. A longitudinal study comparing learning outcomes between traditional dissection and Anatomia-augmented instruction would be necessary to establish the pedagogical impact of the tool.

From these limitations, there are several avenues for future research. The anatomical model library could be extended from the head area to the entire body in order to check the framework's scalability with regard to more complicated structures. The two-user limit of the conflict resolver might be increased by incorporating a third-party user-identification pipeline at the application level, which could in turn allow three or more users to use the system at the same time on larger display surfaces. A formal computational-performance benchmark covering a range of hardware configurations would define the boundaries for deployment. Longitudinal classroom studies that compare Anatomia-augmented instruction with traditional dissection using standardised anatomy assessments would determine whether the usability and interaction benefits reported here lead to better learning outcomes. Lastly, future evaluations of Anatomia could adopt a standardised usability instrument with established reliability. The use of such instruments with a larger and more diverse group of participants, preferably in several different institutions and educational settings, would improve generalisability and enable internal-consistency reporting, cross-study comparison, and more confident interpretation of subjective responses.

VI. Conclusion

This study presents Anatomia, a 3D anatomy visualization application for tabletop display systems that addresses three limitations common to current 3D anatomy visualization tools: the lack of domain-specific gestures, the lack of adaptive content orientation, and the lack of multi-user support on shared surfaces. The summative evaluation conducted with experienced and novice users confirms that all three contributions work together without trading off one for another. The domain-specific gestures defined in the participatory design achieved their highest task performance with minimal guidance within a single session. Basic touchscreen gestures (single-tap, single-finger drag, two-finger pinch) succeeded on the first attempt for both participant groups, and the novel gestures succeeded after a single round of verbal guidance, with the cross-section gesture the only task whose non-medical group required a second observation. The content-alignment mechanism, which pairs automatic rotation driven by radial-menu position with manual rotation and locking, received a median usability score of 0.88 on the alignment task set, with identical performance across medical and non-medical groups (Cliff's $\delta = 0.00$) and 100.0% participant endorsement on the post-session questionnaire. Both performance and self-reported metrics indicate that the mechanism removes the reading-time penalty associated with fixed orientation on shared surfaces without introducing a perceptible usability tax. The software-level conflict resolver achieved a perfect task-success score of 100.0% in two-user interactions (Mdn = 1.00, zero variance across all 13 participants) and 100.0% participant endorsement. It is the first mechanism to support two-user multi-touch interaction without antenna-embedded furniture or external cameras in the domain of medical anatomy education, which is the principal contribution of this work in terms of classroom viability. Three concrete extensions follow from this work. The anatomical model layer can be expanded from the head region to the rest of the body. The conflict resolver's two-user limit can be lifted by integrating a third-party user-identification pipeline. Finally, a longitudinal study comparing learning outcomes between traditional dissection and Anatomia-augmented instruction would be necessary to confirm that the interaction-level gains documented here translate into measurable learning gains over a curriculum term.

Acknowledgment

The authors wish to thank Dr. Sandaruwan and Dr. Keppitiyagama for their guidance and supervision throughout this research. The authors would also like to extend their appreciation to their advisors, Dr. Karunadasa and Mr. Tony Mahadevan,

for their valuable input and feedback. The authors are grateful to the Faculty of Medicine, University of Wayamba, Sri Lanka, for facilitating the observational study and providing access to dissection lessons, and to the medical science students from the University of Colombo and the University of Wayamba who participated in the questionnaires and evaluation sessions. We acknowledge the funding received from the University of Colombo School of Computing through the Research Allocation for Research and Development. This financial assistance greatly contributed to the success of our research endeavor by covering various research-related expenses.

Declaration on Generative AI

Generative AI tools were used to assist with language refinement and clarity during the preparation of this manuscript. All technical content, data collection and analysis, and conclusions were developed and verified by the authors. Generative AI was not used to generate, fabricate, or alter any experimental data, nor to produce original research claims. The authors take full responsibility for the accuracy, originality, and integrity of the work, and confirm that the manuscript reflects their own intellectual contributions throughout.

References

[1] M. Hu, D. Wattchow, and D. De Fontgalland, "From ancient to avant-garde: a review of traditional and modern multimodal approaches to surgical anatomy education," *ANZ Journal of Surgery*, vol. 88, no. 3, pp. 146–151, 9 2017. [Online]. Available: https://doi.org/10.1111/ans.14189

[2] M. Waithaka, V. Kipkorir, J. Munguti, and B. Olabu, "A review of modern anatomy teaching practices," *Anatomy Journal of Africa*, vol. 12, no. 2, pp. 2337–2339, 8 2023. [Online]. Available: https://doi.org/10.4314/aja.v12i2.1

[3] L. M. Parker, "Anatomical dissection: Why are we cutting it out? Dissection in undergraduate teaching," *ANZ Journal of Surgery*, vol. 72, no. 12, pp. 910–912, 12 2002. [Online]. Available: https://doi.org/10.1046/j.1445-2197.2002.02596.x

[4] M. Estai and S. Bunt, "Best teaching practices in anatomy education: A critical review," *Annals of Anatomy - Anatomischer Anzeiger*, vol. 208, pp. 151–157, 3 2016. [Online]. Available: https://doi.org/10.1016/j.aanat.2016.02.010

[5] S. K. Ghosh, "Cadaveric dissection as an educational tool for anatomical sciences in the 21st century," *Anatomical Sciences Education*, vol. 10, no. 3, pp. 286–299, 8 2016. [Online]. Available: https://doi.org/10.1002/ase.1649

[6] E. A. S. T. Edirisinghe, H. D. K. De Silva, K. A. M. T. Devmini, S. D. S. Pathmaperuma, W. M. K. M. Ratnayake, Y. D. M. Jayepeasad, N. D. B. U. Nakandala, M. A. A. Shiraf, L. A. Indunil, D. H. M. Niluka, P. K. K. Madushika, Y. S. Deegodagamage, W. M. C. S. Wijesundara, P. H. Dissanayake, and S. G. Yasawardene, "Integration of traditional and innovative methods in anatomy dissections: A Sri Lankan experience," *South-East Asian Journal of Medical Education*, vol. 16, no. 2, pp. 46–53, 12 2022. [Online]. Available: http://dx.doi.org/10.4038/seajme.v16i2.395

[7] K. Yammine and C. Violato, "A meta-analysis of the educational effectiveness of three-dimensional visualization technologies in teaching anatomy," *Anatomical Sciences Education*, vol. 8, no. 6, pp. 525–538, 12 2014. [Online]. Available: https://doi.org/10.1002/ase.1510

[8] S. A. Azer and S. Azer, "3D Anatomy Models and Impact on Learning: A review of the quality of the literature," *Health Professions Education*, vol. 2, no. 2, pp. 80–98, 5 2016. [Online]. Available: https://doi.org/10.1016/j.hpe.2016.05.002

[9] B. T. Brinke, P. J. Klitsie, R. Timman, J. J. V. Busschbach, J. F. Lange, and G.-J. Kleinrensink, "Anatomy Education and Classroom versus laparoscopic Dissection-Based training," *Academic Medicine*, vol. 89, no. 5, pp. 806–810, 3 2014. [Online]. Available: https://doi.org/10.1097/acm.0000000000000223

[10] T. Murakami, Y. Tajika, H. Ueno, S. Awata, S. Hirasawa, M. Sugimoto, Y. Kominato, Y. Tsushima, K. Endo, and H. Yorifuji, "An integrated teaching method of gross anatomy and computed tomography radiology," *Anatomical Sciences Education*, vol. 7, no. 6, pp. 438–449, 1 2014. [Online]. Available: https://doi.org/10.1002/ase.1430

[11] B. P. Müller-Stich, N. Löb, D. Wald, T. Bruckner, H.-P. Meinzer, M. Kadmon, M. W. Büchler, and L. Fischer, "Regular three-dimensional presentations improve in the identification of surgical liver anatomy – a randomized study," *BMC Medical Education*, vol. 13, no. 1, p. 131, 9 2013. [Online]. Available: https://doi.org/10.1186/1472-6920-13-131

[12] J. E. Sarmiento, "Effects of 3D visualization tables on students' learning outcomes in gross anatomy." [Online]. Available: https://scholarworks.waldenu.edu/dissertations/15738

[13] P. Dietz and D. Leigh, "DiamondTouch," *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 219–226, 11 2001. [Online]. Available: https://doi.org/10.1145/502348.502389

[14] K. C. Dohse, T. Dohse, J. D. Still, and D. J. Parkhurst, "Enhancing Multi-user Interaction with Multi-touch Tabletop Displays Using Hand Tracking," *First International Conference on Advances in Computer-Human Interaction*, pp. 297–302, 1 2008. [Online]. Available: https://doi.org/10.1109/achi.2008.11

[15] A. Raoof, M. Khalil, E. A. Meguid, D. Brzezinski, and M. Dave, "Exploring AACA members' perceptions of integrating technologies in anatomy education," *Clinical Anatomy*, 5 2026. [Online]. Available: https://pure.qub.ac.uk/en/publications/f3712f7a-6d11-4845-933d-a663f257d8cd

[16] A. Koriat and J. Norman, "Reading rotated words." *Journal of Experimental Psychology Human Perception & Performance*, vol. 11, no. 4, pp. 490–508, 1 1985. [Online]. Available: https://doi.org/10.1037/0096-1523.11.4.490

[17] C. Lundstrom, T. Rydell, C. Forsell, A. Persson, and A. Ynnerman, "Multi-Touch Table System for Medical Visualization: Application to Orthopedic Surgery planning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1775–1784, 11 2011. [Online]. Available: https://doi.org/10.1109/tvcg.2011.224

[18] R. C. Aland, H. J. Hugo, A. Battle, R. Donkin, A. McDonald, H. McGowan, J. R. Nealon, H. Ritchie, A. Stirling, M. Tentrisanna, K. Valter, and J. Xiao, "A plethora of choices: an anatomists' practical perspectives for the selection of digital anatomy resources," *Smart Learning Environments*, vol. 10, no. 1, 12 2023. [Online]. Available: https://doi.org/10.1186/s40561-023-00285-3

[19] A. Bartoletti-Stella, V. Gatta, G. A. Mariani, P. Gobbi, M. Falconi, L. Manzoli, I. Faenza, and S. Salucci, "Three-Dimensional virtual anatomy as a new approach for medical student's learning," *International Journal of Environmental Research and Public Health*, vol. 18, no. 24, p. 13247, 12 2021. [Online]. Available: https://doi.org/10.3390/ijerph182413247

[20] A. D. Atlas, "Learn human anatomy in 3D." [Online]. Available: https://anatomy3datlas.com/

[21] C. Anatomy, "Complete Anatomy." [Online]. Available: https://3d4medical.com/

[22] S. K. Motsinger, "Complete anatomy," *Journal of the Medical Library Association JMLA*, vol. 108, no. 1, 1 2020. [Online]. Available: https://doi.org/10.5195/jmla.2019.853

[23] B. Albert and T. Tullis, *Performance Metrics*, 3rd ed. Morgan Kaufmann, 1 2022. [Online]. Available: https://doi.org/10.1016/b978-0-12-415781-1.00004-2

[24] H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 3 1947. [Online]. Available: https://doi.org/10.1214/aoms/1177730491

[25] N. Cliff, "Dominance statistics: Ordinal analyses to answer ordinal questions." *Psychological Bulletin*, vol. 114, no. 3, pp. 494–509, 11 1993. [Online]. Available: https://doi.org/10.1037/0033-2909.114.3.494

[26] J. Romano, J. D. Kromrey, J. Coraggio, and J. Skowronek, "Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys," *Annual meeting of the Florida Association of Institutional Research*, vol. 177, no. 34, 2 2006.

[27] B. Albert and T. Tullis, *Self-Reported Metrics*, 3rd ed. Morgan Kaufmann, 3 2022. [Online]. Available: https://doi.org/10.1016/b978-0-12-818080-8.00005-4

[28] C. Nancarrow and I. Brace, "Saying the "Right Thing": Coping with Social Desirability Bias in Marketing Research." *Bristol Business School Teaching and Research Review*, vol. 3, pp. 1–11, 2000.

[29] S. Criollo-C, A. Guerrero-Arias, Y. M. Arif, A. D. Samala, A. Jaramillo-Alcázar, and S. Luján-Mora, "Usability evaluation of a mobile augmented reality app for PC hardware training: a comparative study in three countries," *Emerging Science Journal*, vol. 9, no. 2, pp. 977–994, 4 2025. [Online]. Available: https://doi.org/10.28991/esj-2025-09-02-024

Does It Generalize? A Cross-Dataset Study of Graph-Based Insider Threat Detection Beyond CERT

QASIM Mohamed Muhanna Alriyami, Mohd Murtadha Bin Mohamad
Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia

***Abstract*—Graph-based insider threat detection methods increasingly report strong performance by modelling relationships among users, hosts, resources and security events. However, many studies still evaluate within a single benchmark family—usually CERT—or report separate within-dataset results, so it is unclear whether the learned representations capture transferable insider behaviour or dataset-specific artefacts. This study presents, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. In this setting, no labelled or unlabelled target observations are used during training. Events are harmonised to a common user-day prediction unit with label-independent event sampling, and in-distribution performance is controlled with both chronological and user-disjoint splits. Within-domain performance of a Random Forest reference model was heterogeneous (user-disjoint PR-AUC from 0.004 on LANL to 0.99 on SPEDIA). Across evaluable targets, its mean in-distribution PR-AUC was 0.542, compared with a mean off-diagonal zero-shot PR-AUC of 0.133, yielding a generalization gap of 0.409. A memory-based temporal graph neural network without explicit time encoding was compared with this reference on two transfers into SPEDIA using paired seed-level bootstrap intervals. Representation effects were source-dependent: the temporal graph model outperformed the Random Forest for CERT r4.2→SPEDIA (ΔPR $= +0.123$, 95% CI $[+0.075, +0.169]$), whereas the Random Forest outperformed the graph model for CERT r5.2→SPEDIA (ΔPR $= -0.031$, 95% CI $[-0.061, -0.012]$). The results therefore do not support a general advantage of graph representations. They show that strong single-dataset performance is insufficient evidence of cross-dataset generalization.**

***Keywords*—Insider threat detection; cross-dataset generalization; zero-shot evaluation; graph neural networks; temporal graphs; self-supervised learning; CERT; SPEDIA; LANL**

I. Introduction

Insider threats remain difficult to detect because malicious, negligent, or compromised users may operate through legitimate accounts, authorised devices, and routine organisational workflows. Unlike many external attacks, insider activity does not necessarily produce an immediate perimeter violation. Its significance may instead emerge through a sequence of individually plausible actions involving authentication, host access, files, removable devices, communications, and other enterprise resources. The difficulty of obtaining suitable insider-threat data, together with confidentiality and privacy constraints on organisational logs, further complicates the development and evaluation of reliable detectors [1].

Machine-learning and deep-learning methods have improved the ability to model these behavioural signals. Recent graph-based approaches are especially promising because enterprise activity can be represented as interactions among users, hosts, resources, and timestamped events. Graph neural networks can propagate relational context between connected entities, while temporal graph models can retain changes in behaviour across ordered interactions. ATHITD uses a temporal heterogeneous graph to represent insider activity, whereas Log2Graph transforms security logs into graph structures for graph-convolutional detection [2], [3]. These developments extend insider threat detection beyond isolated records and fixed aggregated feature vectors.

However, methodological progress has not been matched by equally rigorous evidence of cross-dataset transfer. The synthetic CERT corpus remains the dominant experimental benchmark, and several recent sequence, relational, and graph-based studies continue to evaluate their methods using only CERT-derived data or, like ATHITD and Log2Graph, CERT and LANL without cross-dataset testing [2]–[6]. Reviews also document frequent use of CERT in insider-threat research [7]. CERT provides controlled scenarios, multiple enterprise event sources, and answer-key labels, making it valuable for reproducible model development [1]. Nevertheless, a detector trained and tested within one CERT release may learn scenario-specific frequencies, user-role patterns, event-generation rules, or other corpus regularities that are absent from independently constructed datasets. Strong within-dataset performance therefore does not establish that the learned representation will remain effective in a different organisational or logging environment.

Network intrusion detection has addressed this concern more directly. Cross-dataset studies have demonstrated that models performing strongly on a source benchmark may deteriorate when evaluated on independently collected network traffic [8], [9]. Together, these studies motivate source–target evaluation matrices for comparing within-dataset and cross-dataset performance [8], [9], while Sarhan et al. examine a common feature set for cross-environment generalisability [10]. (A source–target matrix scores every train-on-one, test-on-another dataset pair; a harmonised representation uses one shared feature set for all datasets.) Zero-shot transfer, which applies a trained model to a new dataset without retraining or tuning on it, has also been studied under protocol and dataset shift [11]. Cross-dataset evaluation is consequently treated as a distinct benchmark rather than as an extension of a random train–test split. Insider threat detection has not yet adopted an equivalent evaluation paradigm. To the best of our knowledge, prior graph-based insider-threat studies have not evaluated transfer from CERT to an independently constructed target under a strict zero-shot protocol, and systematic source-to-target benchmarking across independently

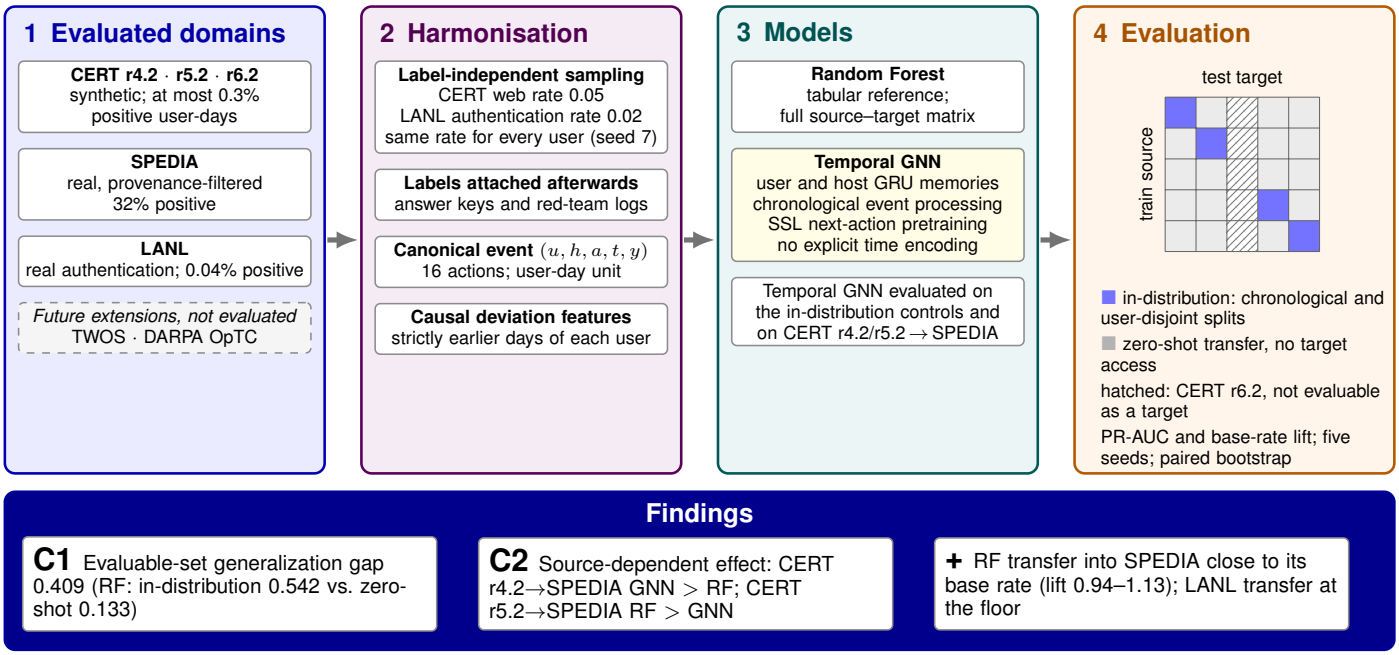


Fig. 1. Overview of the cross-dataset benchmark. Three evaluated domain families are harmonised to a canonical event schema and a user-day prediction unit with label-independent event sampling. A Random Forest reference model is evaluated on the full source-by-target matrix, and a memory-based temporal graph neural network without explicit time encoding is evaluated on the in-distribution controls and on two transfers into SPEDIA. The findings panel summarises the evaluable-set generalization gap and the source-dependent representation comparison.

constructed insider-threat datasets remains limited. It therefore remains unclear whether recent graph models learn transferable insider-threat behaviour or dataset-specific decision rules. This uncertainty is particularly important because CERT, SPEDIA, and LANL differ in data origin, event coverage, insider type, labelling procedure, population, and class prevalence [1], [12], [13].

This study addresses the gap with a *zero-shot cross-dataset generalization* benchmark: a model is trained on a source dataset and evaluated on an independently defined target dataset without access to labelled or unlabelled target observations. Target-assisted domain adaptation is outside the scope of the reported experiments, so no result in this study uses target data during training. The principal contributions are as follows:

- *A strict zero-shot cross-dataset benchmark.* We present, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. It combines explicit target isolation, label-independent event sampling and harmonisation, chronological and user-disjoint in-distribution controls, base-rate lift reporting and an evaluable-set generalization gap. For a Random Forest reference model, mean PR-AUC fell from 0.542 in distribution to 0.133 under zero-shot transfer ($\Delta = 0.409$).
- *A source-dependent comparison of tabular and temporal-graph representations.* Using paired seed-level bootstrap intervals, we compare the reference model with a memory-based temporal graph neural network on two transfers into SPEDIA. The graph model was better for CERT r4.2 as the source, and the tabular model was better for CERT r5.2. Representation advantages therefore did not transfer consistently, even into the same target dataset.

The remainder of this study is organised as follows. Section II reviews insider threat detection, graph-based methods, cross-dataset network intrusion detection, and the selected datasets. Section III defines the datasets, prediction unit, and zero-shot benchmark. Section IV describes the temporal graph construction and detector, Section V describes the experimental protocol, and Section VI reports the findings. Sections VII and VIII discuss threats to validity and conclude the study, respectively. Fig. 1 summarises the overall framework.

II. Related Work

A. Machine Learning and Deep Learning for Insider Threat Detection

Insider threat detection has commonly been formulated as supervised classification, anomaly detection, or behavioural profiling over enterprise activity records. Conventional methods aggregate events into user-level, session-level, or user-day features and apply classifiers or outlier detectors to distinguish suspicious activity from normal behaviour. These methods are computationally practical and can provide useful operational summaries. However, aggregation may obscure the event ordering, timing, and interaction context through which gradual insider behaviour becomes observable.

Deep-learning studies attempt to preserve more of this structure. Recurrent neural networks, autoencoders, attention mechanisms, and transformer-based models can represent behavioural sequences and identify deviations from historical activity. Temporal feature aggregation with attention has been

used to combine activity-log evidence across time [5], while relational multi-sequence modelling has been proposed to capture dependencies among different behavioural sources [6]. LAN learns adaptive neighbourhood relationships for real-time insider threat detection, illustrating the increasing use of relational learning beyond fixed feature aggregation [4]. However, existing machine-learning and sequence-based studies provide limited evidence that their learned behavioural representations remain effective outside the dataset on which they were developed.

B. Graph-Based Insider Threat Detection

Graph-based insider threat detection represents enterprise activity through relationships among users, hosts, resources, and security events. Such representations can preserve interaction structure that is lost when each user-day is treated as an independent feature vector. Static graph convolutional networks aggregate neighbourhood information, while graph attention mechanisms learn the relative importance of connected entities or event types. Temporal graph models additionally retain the order and timing of interactions.

Log2Graph converts security logs into graph structures and applies graph convolution for insider threat detection [3]. Its formulation demonstrates the value of representing log relationships explicitly rather than relying only on flat event aggregates. ATHITD is one of the closest works to the present study because it combines attention with a temporal heterogeneous graph representation for insider threat detection [2]. A subsequent fusion model integrates sequential and graph neural components with an explainability mechanism for user and entity behaviour analytics [14].

ATHITD and the fusion model are particularly relevant architectural comparators, but neither measures cross-dataset transfer. ATHITD reports results on CERT and LANL and the fusion model on CERT, yet neither trains on one insider threat dataset and evaluates on another, independently constructed dataset. Their reported results therefore measure within-dataset discrimination rather than zero-shot cross-dataset transfer. Log2Graph is subject to the same general limitation: it demonstrates a graph-based detector within its development setting but does not establish whether the learned representation transfers across dataset boundaries [3].

The present study does not reimplement these architectures. Instead, it evaluates a memory-based temporal graph detector and a tabular reference model under the same user-day labels, aligned event vocabulary, source–target partitions, and access restrictions. Consequently, existing graph-based detectors have not been compared systematically under a zero-shot source-by-target protocol that separates strong single-dataset performance from transferable graph learning.

C. Cross-Dataset Generalization and Domain Adaptation in Network Intrusion Detection

Network intrusion detection provides the closest established precedent for the present evaluation design. Layeghy and Portmann examined the generalisability of machine-learning network intrusion detectors and showed that strong performance on a development dataset does not necessarily persist under cross-dataset evaluation [8]. Sarhan et al. investigated standardised feature sets intended to improve both generalisability and explainability across network intrusion datasets [10]. Cantone et al. further analysed cross-dataset generalization and the effects of distributional differences between independently collected benchmarks [9].

These studies motivate the use of a source–target matrix rather than a single within-dataset score. A model is trained on one source and evaluated on multiple targets, allowing diagonal in-distribution performance to be separated from off-diagonal transfer. Such evaluation exposes dataset bias that may remain hidden when training and testing records originate from the same collection process.

Domain adaptation addresses a related but different problem. Instead of requiring transfer without target access, adaptation methods use information from the target domain to reduce the distribution mismatch; because the target distribution participates in training, such methods address unsupervised domain adaptation rather than zero-shot generalization. Oladejo and Ahmed apply cross-domain transfer learning to multi-protocol intrusion detection, but their reported experiments are zero-shot, with no target fine-tuning or domain adaptation [11]. The present study likewise reports zero-shot results only.

Although the distinction between source-only cross-dataset generalization and target-assisted unsupervised adaptation is established in network intrusion research, it has not been applied systematically to graph-based insider threat detection.

D. Insider Threat Datasets

The CERT insider-threat corpus was created to support reproducible research when labelled organisational insider incidents were difficult to obtain [1]. It provides synthetic users, hosts, organisational roles, multi-source activity records, and answer-key labels for traitor scenarios. Its controlled structure and accessibility explain its widespread use, but its synthetic generation process may also introduce regularities that are not present in independently collected environments.

TWOS was developed from a gamified competition containing malicious insider behaviour [15]. It includes labelled masquerader and traitor activity spans and therefore differs from CERT in collection setting, scenario construction, and available event fields. These differences make TWOS a natural candidate for extending the present benchmark. It is therefore identified as a target for future work and is not evaluated in this study.

SPEDIA provides security-event data with behavioural descriptions and Wazuh severity information [12]. The release contains both real and synthetic components, including records associated with CERT. A clean CERT-to-SPEDIA comparison therefore requires provenance-aware filtering so that CERT-derived records do not appear in the target evaluation. In this study, every row for which `Decoder_name == 'cert'` is excluded. The audited remainder contains 20,619 real rows, zero occurrences of `dtaa.com`, zero `cert` tokens, and zero CERT-style identifiers. The retained records are drawn only from auditd, PAM, syscheck, and JSON sources.

The LANL cybersecurity dataset contains authentication and other enterprise events collected from a real organisational environment [13]. Its labelled red-team authentication activity differs semantically from CERT traitor scenarios and is more

closely associated with masquerader or lateral-movement behaviour. LANL nevertheless provides a valuable independently collected domain for testing whether models transfer beyond synthetic insider scenarios.

These datasets cannot be made semantically identical through preprocessing. CERT, SPEDIA, and LANL differ in event types, malicious-activity definitions, temporal coverage, and labelling granularity. The present work therefore aligns them at a common user-day prediction level while retaining and reporting their original threat semantics.

What is still missing is a harmonised benchmark that evaluates insider threat detectors across CERT, SPEDIA, and LANL without concealing the substantive differences in their labels and collection processes.

III. Datasets and Problem Formulation

A. Datasets

1) CERT: CERT provides synthetic enterprise activity together with answer-key labels for traitor scenarios [1]. Releases r4.2, r5.2, and r6.2 are used. Events are mapped to user-day intervals, and a user-day is labelled positive when it contains at least one answer-key malicious event. CERT is used as a controlled source domain whose in-distribution performance is measured under both chronological and user-disjoint splits. The term "overfitting" is used only when these controls support that interpretation; otherwise, an off-diagonal decline is described as failure to transfer.

2) SPEDIA: SPEDIA contains security records with descriptive and Wazuh severity information [12]. A user-day is labelled positive when at least one event has a `Description` of "Highly Suspicious" or "Midly Suspicious", or a Wazuh level of at least 8. To prevent CERT-derived overlap, all rows satisfying `Decoder_name == 'cert'` are excluded. The audited real-only subset contains 20,619 rows, zero `dtaa.com` occurrences, zero `cert` tokens, and zero CERT-style identifiers. Its retained sources are limited to auditd, PAM, syscheck, and JSON records.

3) LANL: The LANL dataset contains authentication and related events from a real enterprise environment, together with labelled red-team activity [13]. Each authentication event is mapped to the corresponding user and host within the aligned action vocabulary. A user-day is positive when it contains at least one labelled red-team authentication event. Because these labels principally represent masquerader or lateral-movement activity, LANL is treated as a related but not identical insider-threat domain. Authentication records of machine accounts are removed.

4) Label-independent event sampling: The CERT web logs (14.5–90 GB per release) and the LANL authentication log (about 1.05 billion rows) are too large to process in full, so these two event sources are subsampled. Every CERT web event is retained independently with probability 0.05, and every eligible LANL authentication event is retained independently with probability 0.02, using a fixed pseudo-random seed (7). The retention decision depends only on this random draw; it does not use user identity, the ground-truth insider or red-team rosters, or any label. User-day labels are attached only after the sampled event universe has been fixed. All other CERT event sources are retained in full, and SPEDIA is not subsampled. A labelled user-day can be observed only if at least one of its events is retained; for LANL, 158 of the 176 red-team user-days remain observable after sampling.

5) Evaluable targets: A domain is evaluated as a target, and its in-distribution score is reported, only when it contains at least 50 positive user-days. CERT r6.2 contains 44 positive user-days among 1,394,010 user-days and is therefore used as a training source only. The other four domains meet the criterion.

B. Common Prediction Unit and Graph Domain

Let the benchmark contain K domains,

$$\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}, \tag{1}$$

where, each domain corresponds to one dataset and defines a distribution over timestamped security events. An event is represented as

$$e = (u_e, h_e, a_e, t_e), \tag{2}$$

where, u_e is the user, h_e is the host, a_e is an action from the fixed vocabulary $\mathcal{A}$, and t_e is the event timestamp. All datasets are converted to the same user-day prediction unit. For user u and day d, let $\mathcal{E}_{u,d}$ denote the set of events assigned to that interval. Its binary label is

$$y_{u,d} = \mathbb{I}\left[\exists\, e \in \mathcal{E}_{u,d} \text{ such that } \ell_e = 1\right], \tag{3}$$

where, ℓ_e is the dataset-specific malicious-event or malicious-span indicator. This definition preserves each dataset's documented labelling rule while enforcing a common unit of analysis. The user-day is the unit of labelling, prediction, and evaluation. It is not the input resolution of the temporal model. The temporal graph detector consumes individual timestamped events in chronological order, so event ordering and within-day temporal structure enter its memory state; the aggregated user-day features are used alongside that memory. Daily labels nevertheless have limitations. They cannot measure detection latency within a day, event-level detection behaviour, or differences between sessions. Event-, session-, host-, or process-level formulations should therefore be evaluated where the datasets support them.

Each domain is represented as a heterogeneous temporal graph

$$\mathcal{G}_i = \left(\mathcal{V}_i^U, \mathcal{V}_i^H, \mathcal{E}_i\right), \tag{4}$$

where, $\mathcal{V}_i^U$ and $\mathcal{V}_i^H$ are user and host nodes, respectively, and $\mathcal{E}_i$ contains action-typed timestamped edges. Dataset-specific raw events are mapped to the fixed action vocabulary before model fitting. Continuous feature normalisation is fitted using the permitted training data only.

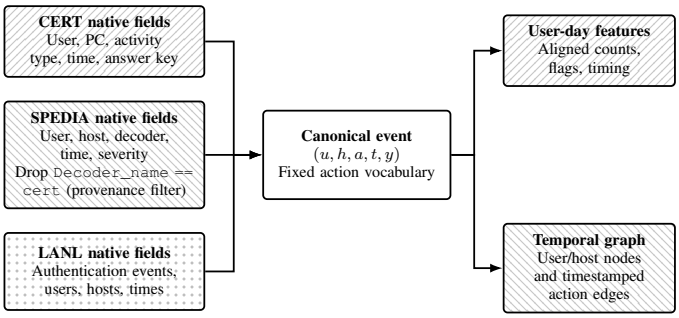


Fig. 2. Data harmonisation pipeline.

Fig. 2 illustrates the harmonisation pipeline.

C. Zero-Shot Cross-Dataset Generalization

The primary benchmark measures zero-shot cross-dataset generalization. For source domain i, a detector f_i^{ZS} is trained using only labelled source-domain training data:

$$f_i^{\mathrm{ZS}} = \mathrm{Train}\left(\mathcal{D}_i^L\right). \tag{5}$$

No labelled or unlabelled target events, target feature statistics, target validation results, or target-domain model-selection signals are available during training.

For source i and target j, the matrix entry is

$$M_{ij} = \text{PR-AUC}\left(f_i^{\mathrm{ZS}}, \mathcal{D}_j^{\mathrm{test}}\right). \tag{6}$$

The diagonal entries represent in-distribution evaluation, whereas $i \neq j$ denotes zero-shot transfer. In-distribution scores are estimated under both a chronological split and a user-disjoint split (Section V). The complete source-by-target matrix is computed for the Random Forest reference model, and the temporal graph detector is evaluated on the in-distribution controls and on two transfers into SPEDIA (Section VI).

Let S denote the set of source domains, $T \subseteq S$ the set of evaluable target domains, and

$$P = \{(i, j) : i \in S,\ j \in T,\ i \neq j\} \tag{7}$$

the set of evaluable transfer cells. The generalization gap is defined as

$$\Delta = \frac{1}{|T|} \sum_{j \in T} M_{jj} - \frac{1}{|P|} \sum_{(i,j) \in P} M_{ij}. \tag{8}$$

The first term is the uniformly weighted mean of the evaluable in-distribution cells, and the second is the uniformly weighted mean of the evaluable zero-shot cells. A larger positive value indicates a greater decline from in-distribution performance to zero-shot transfer. CERT r6.2 remains available as a training source but is excluded from T because its target evaluation does not meet the positive-count criterion; with five sources and four evaluable targets, $|P| = 16$. The in-distribution cells in Eq. (8) use the user-disjoint split, because the chronological split leaves no LANL positives in the test period.

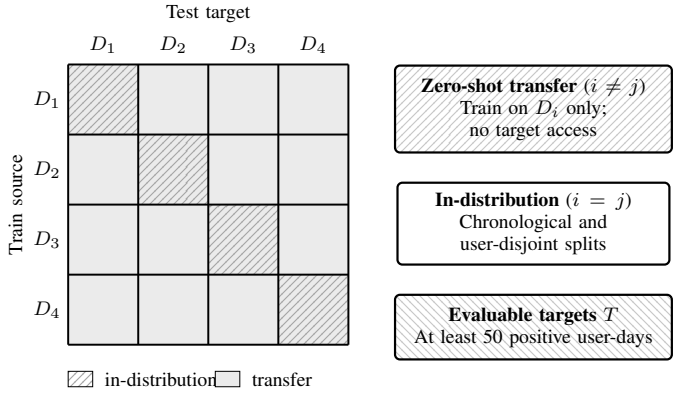


Fig. 3. Zero-shot evaluation protocol. Rows are training sources and columns are test targets; only evaluable targets enter the generalization gap.

The evaluation protocol is summarised in Fig. 3.

IV. Methodology

A. Heterogeneous Temporal Graph Construction

Each dataset is transformed into a heterogeneous temporal graph containing user and host nodes. Every security event creates a directed, action-typed edge between its associated user and host:

$$e = (u_e, h_e, a_e, t_e), \qquad a_e \in \mathcal{A}. \tag{9}$$

The vocabulary $\mathcal{A}$ contains sixteen canonical actions: logon, logoff, auth, file_read, file_write, file_delete, usb_connect, usb_disconnect, http, email_send, email_recv, cmd_exec, ssh, ftp, process_start, and unknown. It was chosen to cover the activity types that the evaluated sources record in comparable form, namely session, authentication, file, removable-device, web, e-mail, and command events, without relying on dataset-specific content. The mapping from raw log fields to $\mathcal{A}$ is a static rule table in the data loaders. It does not use labels or model outputs, and the same tables were applied in all reported experiments. Dataset-specific event names that express the same behavioural operation are mapped to a shared action, whereas unmatched events are assigned to the reserved unknown action rather than discarded. Table I lists the rules, and Table II reports the resulting event distribution. No event in any domain was assigned to the unknown action. The distributions nevertheless differ markedly: LANL occupies only three actions, SPEDIA is dominated by command execution, and CERT is dominated by e-mail and web activity. The mapping therefore provides a common representation, but it cannot make the source domains semantically identical.

In addition to event edges, each user-day has an aligned feature vector

$$\mathbf{x}_{u,d} \in \mathbb{R}^F, \tag{10}$$

TABLE I. CANONICAL ACTION MAPPING. CERT R4.2 HAS NO ACTIVITY COLUMN FOR FILE AND E-MAIL EVENTS, SO ALL ITS FILE EVENTS MAP TO FILE_WRITE AND ALL ITS E-MAIL EVENTS TO EMAIL_SEND. CERT WEB EVENTS ARE SAMPLED INDEPENDENTLY WITH PROBABILITY 0.05 FOR EVERY USER, AND ELIGIBLE LANL AUTHENTICATION EVENTS WITH PROBABILITY 0.02 FOR EVERY USER (FIXED SEED 7; MACHINE ACCOUNTS REMOVED). GROUND-TRUTH ROSTERS ARE NOT USED IN THE SAMPLING DECISION, AND LABELS ARE ATTACHED ONLY AFTER SAMPLING. THE SSH, FTP, AND PROCESS_START ACTIONS ARE RESERVED AND ARE NOT PRODUCED BY ANY EVALUATED LOADER

Canonical action	CERT (r4.2/r5.2/r6.2)	SPEDIA (real-only)	LANL (auth.txt)
logon	logon.csv: Logon	Activity = session, other actions	Orientation LogOn
logoff	logon.csv: Logoff	Activity = session, action contains “logoff”	Orientation LogOff
auth	–	Activity = session, action contains “fail”	All other orientations
file_read	file.csv: File Open (r5.2/r6.2)	Activity = file, other actions	–
file_write	file.csv: File Copy, File Write; all r4.2 file events	Activity = file, action contains “modif” or “added”	–
file_delete	file.csv: File Delete	Activity = file, action contains “delet”	–
usb_connect	device.csv: Connect	Activity = device, other actions	–
usb_disconnect	device.csv: Disconnect	Activity = device, action contains “disconnect”	–
http	http.csv (sampled)	Activity = http	–
email_send	email.csv: Send; all r4.2 e-mail events	Activity = email	–
email_recv	email.csv: View, Receive (r5.2/r6.2)	–	–
cmd_exec	–	Activity = command	–
unknown	Any unmatched token	Any other activity	–

TABLE II. DISTRIBUTION OF RETAINED EVENTS OVER CANONICAL ACTIONS AFTER LABEL-INDEPENDENT SAMPLING: COUNT (PERCENTAGE OF THE DOMAIN’S EVENTS). ONLY ACTIONS OBSERVED IN AT LEAST ONE DOMAIN ARE LISTED; SSH, FTP, AND PROCESS_START HAVE ZERO EVENTS IN EVERY DOMAIN. THE LAST ROW GIVES USER-DAYS AND, IN PARENTHESES, POSITIVE USER-DAYS

Action	CERT r4.2	CERT r5.2	CERT r6.2	SPEDIA	LANL
logon	470,591 (8.2%)	995,082 (4.2%)	1,948,933 (8.1%)	3,031 (14.7%)	3,786,729 (45.3%)
logoff	384,268 (6.7%)	814,988 (3.4%)	1,581,352 (6.6%)	2,038 (9.9%)	3,048,812 (36.4%)
auth	0 (0.0%)	0 (0.0%)	0 (0.0%)	330 (1.6%)	1,529,753 (18.3%)
file_read	0 (0.0%)	483,655 (2.0%)	657,500 (2.7%)	0 (0.0%)	0 (0.0%)
file_write	445,581 (7.7%)	403,966 (1.7%)	976,584 (4.1%)	4,027 (19.5%)	0 (0.0%)
file_delete	0 (0.0%)	0 (0.0%)	380,799 (1.6%)	712 (3.5%)	0 (0.0%)
usb_connect	203,339 (3.5%)	419,734 (1.8%)	778,354 (3.3%)	0 (0.0%)	0 (0.0%)
usb_disconnect	202,041 (3.5%)	417,250 (1.7%)	773,474 (3.2%)	0 (0.0%)	0 (0.0%)
http	1,420,426 (24.7%)	2,947,136 (12.4%)	5,850,511 (24.4%)	367 (1.8%)	0 (0.0%)
email_send	2,629,979 (45.7%)	5,555,187 (23.3%)	3,832,466 (16.0%)	0 (0.0%)	0 (0.0%)
email_recv	0 (0.0%)	11,806,388 (49.5%)	7,162,491 (29.9%)	0 (0.0%)	0 (0.0%)
cmd_exec	0 (0.0%)	0 (0.0%)	0 (0.0%)	10,114 (49.1%)	0 (0.0%)
unknown	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Total events	5,756,225	23,843,386	23,942,464	20,619	8,365,294
User-days (positive)	330,452 (966)	693,649 (1,307)	1,394,010 (44)	256 (82)	389,323 (158)

whose dimensions have the same operational meaning across all datasets. The aligned features contain counts, binary activity indicators, timing variables, and other quantities that can be computed consistently from the common event ontology. Dataset identifiers, raw user names, host names, and source-specific record identifiers are excluded.

After processing all events up to the end of day d, the user-day representation is

$$\mathbf{r}_{u,d} = \left[\mathbf{s}_u(t_d^{\text{end}}) \,\|\, \mathbf{x}_{u,d}\right], \tag{11}$$

where, $\mathbf{s}_u(t_d^{\text{end}})$ is the temporal memory of user u and $\|$ denotes concatenation. A prediction head produces

$$\hat{y}_{u,d} = \sigma\left(\mathbf{W}_y \mathbf{r}_{u,d} + \mathbf{b}_y\right). \tag{12}$$

Fig. 4 shows the architecture.

B. Per-User Deviation Features

In line with user-centred and behavioural-baseline approaches [2], [16], insider activity is modelled relative to each user’s own history rather than solely through absolute activity counts. The same number of logons, file operations, or host accesses may therefore be routine for one user but unusual for another. To preserve temporal validity, the behavioural baseline is constructed causally from strictly earlier observations only. The transformation is also dataset-agnostic: it uses aligned numerical activity features and excludes dataset-specific tokens, URLs, email content, identifiers, or other source-dependent fields.

For user u, day d, and base feature x, the causal deviation feature is

$$\text{dev}_x(u,d) = \frac{x_{u,d} - \mu_{u,<d}}{\sigma_{u,<d} + \epsilon}, \tag{13}$$

where, $\mu_{u,<d}$ and $\sigma_{u,<d}$ are the mean and standard deviation of feature x over the strictly earlier days of user u, and

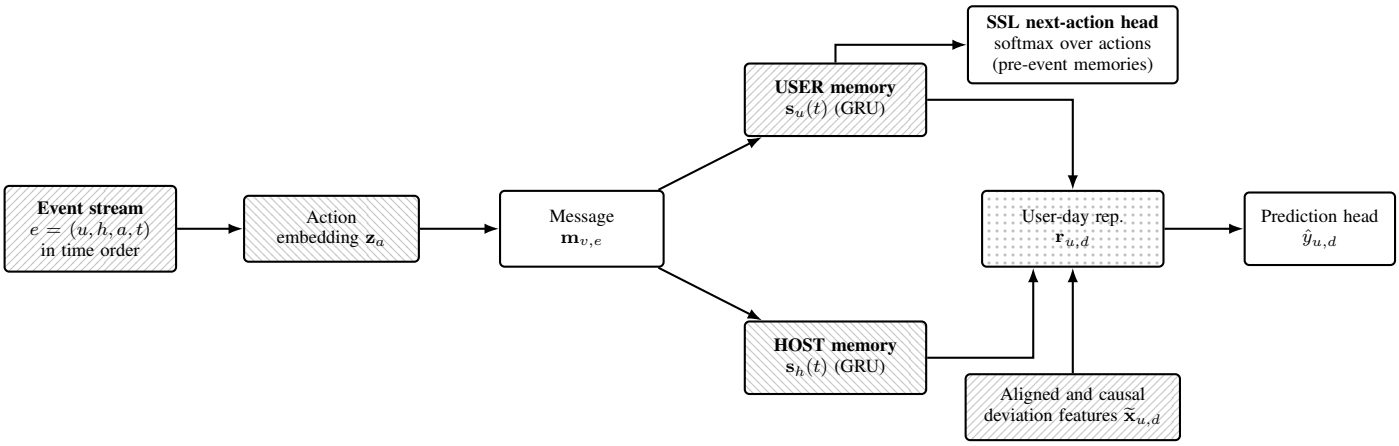


Fig. 4. Temporal graph detector used in the reported experiments. Events update user and host memories in chronological order; no explicit time encoding is used.

$\epsilon > 0$ prevents division by zero. Formally,

$$\mu_{u,<d} = \frac{1}{|\mathcal{H}_{u,d}|} \sum_{d' \in \mathcal{H}_{u,d}} x_{u,d'}, \qquad \mathcal{H}_{u,d} = \{\, d' \mid d' < d \,\}, \tag{14}$$

with $\sigma_{u,<d}$ computed over the same historical set $\mathcal{H}_{u,d}$. When $\mathcal{H}_{u,d}$ is empty, corresponding to the first observed day of user u, $\mathrm{dev}_x(u,d)$ is set to zero. The original base counts are retained, and the causal deviation features are appended. If the aligned base vector is $\mathbf{x}_{u,d} \in \mathbb{R}^F$, the resulting user-day feature vector is

$$\widetilde{\mathbf{x}}_{u,d} = [\, \mathbf{x}_{u,d} \,\|\, \mathrm{dev}(\mathbf{x}_{u,d}) \,] \in \mathbb{R}^{2F}. \tag{15}$$

Because the same aligned base features and the same causal transformation are applied in every domain, the resulting feature space is identical across CERT, SPEDIA, and LANL.

1) Leakage controls: The following rules determine what information can reach each prediction. First, event sampling is label-independent and precedes label attachment (Section III), so the retained event volume of a user does not depend on whether that user is an insider. Second, each deviation value uses an expanding mean and standard deviation over the same user's strictly earlier days; the current day is excluded by a one-day shift, and a user's first observed day receives zero. Only activity counts and timing variables enter this computation, so labels never affect the history. Third, the features are computed once over each dataset's chronological event stream. Under the chronological split, a test day may therefore draw on the same user's training-period days, which precede it in time, but never on later days. Under the user-disjoint split, train and test users do not overlap, so a test user's history consists only of that user's own events. Fourth, during cross-dataset inference, target deviations are computed only from the same target user's earlier, unlabelled target events. Fifth, the temporal-graph memory is reset before each training epoch, before validation, and before scoring, so memory state is not carried from the training partition into the test partition or from the source into the target. Sixth, the temporal graph detector uses no timestamp statistic of the scored stream, because it has no explicit time encoding (Section IV). Finally, standardisation statistics are fitted on the source training users only, excluding the validation users, and are applied unchanged to validation, test, and target data. Consequently, no feature value depends on future events, on labels, or on statistics of a test or target partition, and target labels do not enter model training or preprocessing.

2) Shared feature-space limitation: The shared, content-free feature space is a deliberate cross-dataset fairness constraint. It prevents models from exploiting dataset-specific tokens, URLs, email content, identifiers, or logging conventions that could artificially inflate within-dataset performance and reintroduce the artefacts examined by the benchmark. This restriction may reduce sensitivity to scenarios whose malicious meaning depends on content, such as particular URLs, email text, file names, or message semantics. Per-user deviation features partly mitigate this limitation by representing unusual behaviour relative to each user's own history, but cannot recover information absent from the aligned feature space.

C. Memory-Based Temporal Graph Backbone

The temporal backbone maintains a state vector $\mathbf{s}_v(t)$ for every user or host node v. For event $e = (u_e, h_e, a_e, t_e)$, a message is constructed separately for each endpoint $v \in \{u_e, h_e\}$:

$$\mathbf{m}_{v,e} = \mathbf{W}_m \left[\mathbf{s}_v(t_e^-) \,\|\, \mathbf{z}_{a_e} \right], \tag{16}$$

where, $\mathbf{z}_{a_e}$ is the embedding of action a_e, $\mathbf{s}_v(t_e^-)$ is the memory of node v immediately before event e, and separate projections $\mathbf{W}_m$ are used for user and host endpoints. Temporal graph models often add explicit temporal information, e.g., elapsed-time terms or functional time encodings [17], [18]. Here, such an encoding would need timestamp normalisation, and normalising with the scored target stream's first and last timestamps would use target-derived statistics. The reported configuration therefore uses no explicit time encoding. Temporal information enters the model through the chronological order in which events update the memories.

The node memory is updated as

$$\mathbf{s}_v(t_e) = \mathrm{GRU}\left(\mathbf{m}_{v,e}, \mathbf{s}_v(t_e^-)\right). \tag{17}$$

Events are processed in chronological order. Consequently, the state used to predict event e contains only information available before t_e, preventing future-event leakage. Host memories allow activity on a shared machine to influence subsequent user representations, while user memories retain behavioural continuity across events and days.

D. Causal Self-Supervised Next-Action Prediction

The primary self-supervised task predicts the action associated with the next chronological event from the pre-event memories of its user and host. Let

$$\mathbf{g}\left(\mathbf{s}_{u_e}(t_e^-), \mathbf{s}_{h_e}(t_e^-)\right) = \mathrm{softmax}\left(\mathbf{W}_g\left[\mathbf{s}_{u_e}(t_e^-) \,\|\, \mathbf{s}_{h_e}(t_e^-)\right] + \mathbf{b}_g\right) \tag{18}$$

be a probability distribution over the $A = |\mathcal{A}|$ action classes. The self-supervised loss is

$$\mathcal{L}_{\mathrm{ssl}} = -\sum_{e \in \mathcal{E}_{\mathrm{train}}} \log g_{a_e}\left(\mathbf{s}_{u_e}(t_e^-), \mathbf{s}_{h_e}(t_e^-)\right). \tag{19}$$

The loss is computed over all permitted training events rather than over a masked subset. The use of pre-event memories makes the objective causal because action a_e is not inserted into the state before its prediction. Self-supervised pretraining is conducted without malicious labels and is followed by supervised source-domain optimisation. A masked-action reconstruction variant was not evaluated in this study.

E. Supervised User-Day Detection

For labelled source-domain user-days, the prediction loss is binary cross-entropy:

$$\mathcal{L}_y = -\sum_{(u,d) \in \mathcal{S}^L} \left[y_{u,d} \log \hat{y}_{u,d} + (1 - y_{u,d}) \log\left(1 - \hat{y}_{u,d}\right)\right]. \tag{20}$$

Any weighting or sampling used to address class imbalance is estimated from source-domain training data only. Target labels are never used to tune the classifier or its operating threshold. For zero-shot experiments, all components are trained with source data only. The temporal encoder is trained with the self-supervised objective and then frozen, after which the projection layer and prediction head are trained on labelled source user-days. Early stopping uses the validation PR-AUC of a held-out 15% of the source training users. The resulting model is frozen before evaluation on a target domain.

TABLE III. CONFIGURATION OF THE REPORTED EXPERIMENTS. WHERE TWO VALUES ARE GIVEN, THE FIRST APPLIES TO CERT SOURCES AND THE SECOND TO THE SPEDIA SOURCE

Item	Setting
Seeds	0–4 (splits, validation users, initialisation)
Event sampling	CERT web $p = 0.05$ and LANL authentication $p = 0.02$ for every user; seed 7; labels attached after sampling
Random Forest	300 trees; balanced-subsample class weights; unscaled features
Temporal memory	64-dimensional user and host GRU memories, updated in chronological event order
Embeddings	16-dimensional action embedding; no explicit time encoding
Readout	Linear–ReLU projection; MLP 128–64–1; dropout 0.1
Optimiser	AdamW; learning rate 3×10^{-4}; weight decay 10^{-4}; cosine schedule; gradient-norm clipping at 1.0
Epochs	at most 60 / 80; self-supervised pretraining 1 / 2 epochs
Batching	chronological edge batches of 16,384 / 8,192 events; one supervised update per day
Early stopping	validation PR-AUC on 15% of source training users; patience 8 / 12; minimum 12 / 20 epochs
Class weighting	positive weight $n_{\mathrm{neg}}/n_{\mathrm{pos}}$, capped at 100
Determinism	deterministic PyTorch algorithms; deterministic cuDNN, benchmarking off; CUBLAS workspace `:4096:8`; hash seed 0
Bootstrap	5,000 resamples of paired per-seed differences; 95% percentile interval
Software	Python 3.11.15; scikit-learn 1.9.0; PyTorch 2.12.1 with CUDA 13.0; NumPy 2.4.4; pandas 3.0.3; SciPy 1.17.1
Hardware	NVIDIA GeForce RTX 5070 Laptop GPU (8 GB); 31 GB RAM

V. EXPERIMENTAL SETUP

A. Implementation Details

The pipeline is implemented in Python. The Random Forest uses scikit-learn, and the temporal graph detector uses PyTorch. Table III lists the configuration of both models. Class imbalance is addressed through positive-class weighting estimated only from source training data. The temporal graph detector standardises base features on the source training users and clips them and the causal deviation features (Section IV-B) to $[-8, 8]$; the Random Forest uses unscaled features. Every reported configuration was run with five random seeds (0–4); GPU training was deterministic. Tables IV and V report seed means, and paired seed-level bootstrap confidence intervals are reported for the two model comparisons in Table VI.

B. Cross-Dataset Protocol

The protocol trains on one source domain and evaluates on target domains without target access. All preprocessing, normalisation, feature selection, threshold selection, and model selection follow the access restrictions defined in Section III. Off-diagonal cells train on the complete source domain and score the complete target domain. For every evaluable domain, the in-distribution controls comprise a chronological split and a user-disjoint split. The chronological split places the earliest 70% of the timeline in training and the remainder in testing; the cut is taken at the 0.7 quantile of user-days for the Random Forest and of event timestamps for the temporal graph detector, so users can appear on both sides. The user-disjoint split assigns 70% of users to training and the remaining users to testing, stratified by whether a user has any positive user-day.

These controls test whether a high diagonal score depends on temporal proximity or on repeated users.

C. Compared Models

The Random Forest is the tabular reference model. It is evaluated on the complete source-by-target matrix and on both in-distribution splits. The temporal graph detector of Section IV is evaluated on the in-distribution controls of CERT r4.2, CERT r5.2, and SPEDIA and on the transfers from CERT r4.2 and CERT r5.2 into SPEDIA. Its complete transfer matrix was not computed because of its computational cost. Both models use the same user-day labels, aligned features, source–target partitions, and access restrictions.

D. Evaluation Metrics

PR-AUC (average precision) is the headline metric because positive user-days are rare. Because PR-AUC depends on prevalence, each value is also reported as base-rate lift, that is, PR-AUC divided by the positive rate of the evaluated set. Model differences are summarised by the mean paired difference across the five seeds, with a 95% percentile bootstrap interval from 5,000 resamples of the per-seed differences. Because the temporal graph detector was evaluated on two transfer cells only, no aggregate significance test across the transfer matrix is reported. The generalization gap in Eq. (8) is reported as a transparency statistic. Threshold-dependent operating-point metrics, such as the detection rate at a fixed false-positive rate or precision at k, are not used as evidence: the implementation derives their thresholds from the evaluated set, which would not be available in deployment.

VI. Results and Discussion

All results use the label-independent sampling and the zero-shot protocol defined in Sections III–V, and all values are means over five seeds. This section first reports the in-distribution controls, then the Random Forest transfer matrix and the generalization gap, and finally the paired comparison between the temporal graph detector and the Random Forest.

A. In-Distribution Controls

Table IV shows that within-domain performance was heterogeneous. Under the user-disjoint split, the Random Forest reached PR-AUC values of 0.550 on CERT r4.2, 0.625 on CERT r5.2, and 0.989 on SPEDIA, but only 0.004 on LANL, whose test base rate was about 0.0004 (lift 9.0×). On both CERT releases, the chronological split gave higher scores than the user-disjoint split, for the Random Forest (0.666 and 0.815) and for the temporal graph detector (0.621 and 0.755). This pattern is consistent with a benefit from observing the same users during training. On SPEDIA, the Random Forest scored highly under both splits (0.989 and 0.992), whereas the temporal graph detector scored 0.807 under the user-disjoint split and 0.972 under the chronological split. The temporal graph detector scored below the Random Forest in every in-distribution cell. The chronological LANL split was not evaluable because its test period contained no positive user-days, and CERT r6.2 does not meet the evaluability criterion.

TABLE IV. In-Distribution PR-AUC (Mean Over Five Seeds). "User-Disjoint" and "Chronological" Denote the Split; "Temporal-GNN" Denotes the Model. n/e: Not Evaluable (the LANL Chronological Test Period Contains no Positive User-Days; CERT r6.2 has 44 Positive User-Days, below the Criterion of 50). n/r: Not Run. User-Disjoint Test Base Rates are about 0.0027 (CERT r4.2), 0.0020 (CERT r5.2), 0.41 (SPEDIA), and 0.0004 (LANL)

	Random Forest		Temporal-GNN	
Domain	User-disjoint	Chronological	User-disjoint	Chronological
CERT r4.2	0.5497	0.6660	0.4443	0.6211
CERT r5.2	0.6254	0.8150	0.4776	0.7554
SPEDIA	0.9895	0.9924	0.8072	0.9720
LANL	0.0036	n/e	n/r	n/e
CERT r6.2	n/e	n/e	n/e	n/e

B. Zero-Shot Cross-Dataset Transfer (C1)

Table V and Fig. 5 report the Random Forest transfer matrix. Transfer performance was substantially lower than in-distribution performance on average. Across the four evaluable targets, the mean in-distribution PR-AUC was 0.542 and the mean of the 16 evaluable zero-shot cells was 0.133, giving a generalization gap of $\Delta = 0.409$ (Eq. (8)). The decline was not uniform. Transfers between CERT releases retained part of the signal relative to the target base rate: these cells reached PR-AUC values between 0.015 and 0.390, with lifts between $5\times$ and $133\times$, although each remained below the in-distribution value of its target. In contrast, transfer into SPEDIA gave PR-AUC values between 0.302 and 0.363 against a target base rate of 0.32, corresponding to lifts of only 0.94–$1.13\times$; the Random Forest therefore ranked SPEDIA user-days close to the level expected by chance. Transfers into LANL (PR-AUC ≤ 0.0010; lift 1.1–$2.4\times$) and from LANL or SPEDIA into CERT (PR-AUC ≤ 0.0088; lift 1.0–$3.0\times$) remained near the performance floor. Because the aggregate gap combines targets whose base rates differ by about three orders of magnitude, it is reported together with the per-cell values and lifts.

TABLE V. Zero-Shot Transfer PR-AUC of the Random Forest Reference Model (Base-Rate Lift in Parentheses); Rows = Train Source, Columns = Test Target. The Diagonal is the User-Disjoint In-Distribution Score (Table IV). CERT r6.2 is a Source Only; n/e = Not Evaluable as a Target

Train ↓ / Test →	Target				
	r4.2	r5.2	r6.2	SPEDIA	LANL
r4.2	0.550 (207×)	0.250 (133×)	n/e	0.315 (0.98×)	0.0008 (2.0×)
r5.2	0.390 (133×)	0.625 (317×)	n/e	0.363 (1.1×)	0.0010 (2.4×)
r6.2	0.015 (5.1×)	0.135 (72×)	–	0.302 (0.94×)	0.0004 (1.1×)
SPEDIA	0.0088 (3.0×)	0.0037 (1.9×)	n/e	0.989 (2.7×)	0.0008 (1.9×)
LANL	0.0029 (0.99×)	0.0019 (1.0×)	n/e	0.341 (1.1×)	0.0036 (9.0×)

C. Temporal Graph vs. Tabular Representations (C2)

Table VI compares the temporal graph detector with the Random Forest on the two transfers into SPEDIA, using paired seed-level bootstrap intervals. The representation effect was source-dependent. For CERT r4.2→SPEDIA, the temporal graph detector reached a mean PR-AUC of 0.438 (lift $1.37\times$), compared with 0.315 for the Random Forest, a paired difference of $+0.123$ (95% CI $[+0.075, +0.169]$); all five

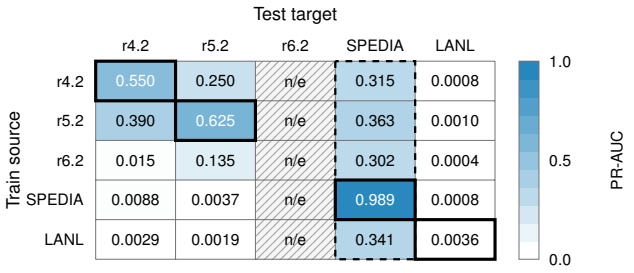


Fig. 5. Zero-shot cross-dataset transfer of the Random Forest (PR-AUC, mean of five seeds). Rows are training sources and columns are test targets; r4.2, r5.2, and r6.2 denote CERT releases. Outlined cells are user-disjoint in-distribution scores. The dashed SPEDIA column has a base rate of 0.32, so its transfer values of 0.30–0.36 are close to chance. CERT r6.2 is not evaluable as a target (hatched) and is excluded from the generalization gap; it is retained as a training source.

seeds favoured the graph model. For CERT r5.2→SPEDIA, the Random Forest reached 0.363 and the temporal graph detector 0.332 (lift $1.04\times$), a paired difference of -0.031 (95% CI $[-0.061, -0.012]$); all five seeds favoured the Random Forest. The graph advantage observed from CERT r4.2 was therefore not reproduced from CERT r5.2, although both sources come from the same synthetic generator and the target is identical. Because the temporal graph detector was not evaluated on the remaining transfer cells, these two comparisons do not support a general statement about graph and tabular representations across the benchmark.

TABLE VI. Paired Comparison of the Temporal-GNN (No Explicit Time Encoding) and the Random Forest on Transfers into SPEDIA (Target Base Rate 0.32). ΔPR = PR-AUC(Temporal-GNN) − PR-AUC(RF), Averaged Over Five Paired Seeds; the 95% CI is a Percentile Bootstrap Over the Paired Per-Seed Differences (5,000 Resamples). The Interpretation Names the Model Favoured by an Interval that Excludes Zero

Source → Target	RF	Temporal-GNN	ΔPR	95% CI	Interpretation
CERT r4.2 → SPEDIA	0.315	0.438	$+0.123$	$[+0.075, +0.169]$	GNN supported
CERT r5.2 → SPEDIA	0.363	0.332	-0.031	$[-0.061, -0.012]$	RF supported

D. Discussion

The results show that strong single-dataset performance is not evidence of cross-dataset transfer. Even where in-distribution scores were high, as for SPEDIA, zero-shot transfer from the other domains into the same target ranked positive user-days close to chance for the Random Forest. The only transfer directions with substantial base-rate lift were those between CERT releases produced by the same simulator. The in-distribution controls add a further caution. On both CERT releases, chronological splits produced higher PR-AUC than user-disjoint splits, so a within-dataset evaluation that places the same users in training and testing can overstate performance on unseen users.

The paired comparison indicates that representation effects depend on the source. The temporal graph detector improved on the tabular reference when trained on CERT r4.2 but not when trained on CERT r5.2, and it scored below the reference in every in-distribution control. A graph representation can therefore help for a particular source–target pair without providing a consistent advantage, even when the target is fixed. Because the complete graph transfer matrix was not computed, the benchmark does not establish whether graph representations are generally better or worse than tabular ones.

Transfer involving LANL remained near the performance floor in both directions. Three non-exclusive explanations are consistent with the evidence. First, the threat semantics differ: LANL positives mark red-team authentication and lateral-movement activity, whereas CERT positives mark synthetic traitor scenarios and SPEDIA positives follow descriptive and Wazuh severity rules. Second, the shared representation covers LANL only partially. All 8.37 million sampled LANL events map to three of the sixteen canonical actions (logon 45.3%, logoff 36.4%, and auth 18.3%; Table II). Authentication type, logon type, and outcome are therefore collapsed, and the file, device, web, and e-mail actions that dominate CERT are absent. Third, prevalence differs by orders of magnitude: the positive user-day rate is about 0.0004 for LANL (158 of 389,323 user-days), compared with 0.32 for SPEDIA and between about 0.00003 and 0.003 for the CERT releases. The present benchmark cannot determine how much of the LANL failure is attributable to each mechanism, and none of them is claimed to be the sole cause.

Finally, the reported results concern ranking and discrimination transfer rather than probability calibration transfer. PR-AUC measures how well positive user-days are ranked above negative ones under class imbalance. It does not show that predicted probabilities keep the same meaning in a new domain, that source confidence scores remain calibrated on the target, or that a fixed operating threshold preserves its false-positive rate. No target recalibration was performed, and calibration metrics such as the Brier score or expected calibration error were not computed. Calibration transfer and operating-point stability therefore remain open questions for deployment.

VII. Threats to Validity

The datasets differ in collection method, event coverage, label semantics, and insider type. CERT contains synthetic traitor scenarios, LANL labels red-team authentication activity, and SPEDIA uses descriptive and Wazuh severity rules. Mapping these sources to a common user-day label supports comparison but cannot make their underlying threat concepts identical. The study therefore interprets cross-dataset results as transfer across related security domains rather than as proof of universal insider-threat detection.

SPEDIA contains CERT-derived records in its original release. This risk is mitigated by excluding all rows for which `Decoder_name == 'cert'` and auditing the remainder for CERT-related tokens and identifiers. The audit supports a clean comparison, but it cannot establish that every retained behavioural pattern is statistically independent of all external data-generation influences.

The fixed action vocabulary and aligned feature set may omit useful dataset-specific information. Conversely, retaining source-specific identifiers or unsupported fields could allow models to recognise the source dataset rather than the behaviour. The study therefore reports the mapping rules

and excludes direct dataset identifiers. No alternative action groupings were evaluated. Because no event in any domain was assigned to the unknown action, retaining or excluding unknown events does not change the reported results; however, sensitivity to alternative groupings, such as separating authentication subtypes in LANL, remains untested. The benchmark compares one tabular and one temporal graph model; other detector families were not evaluated under the final protocol, and the temporal graph detector was evaluated on two transfer cells only.

LANL red-team activity is concentrated within a limited sub-window of the collection period. A chronological split therefore leaves no positives in the test period, so the LANL in-distribution diagonal is reported under the user-disjoint split only. LANL timestamps are also expressed on a relative clock, so time-of-day features are interpreted through the per-user deviation transformation rather than as absolute working hours. The generalization gap assigns uniform weight to all diagonal and off-diagonal cells. This keeps the statistic transparent, but it does not account for differences in target size, prevalence, or label uncertainty. Because target prevalence ranges over roughly three orders of magnitude across domains, per-cell PR-AUC and base-rate lift are reported alongside the aggregate gap, and CERT r6.2 (44 positive user-days) is excluded from the evaluable targets.

The confidence intervals are percentile bootstrap intervals over five paired seeds. They capture variation across seeds but not sampling variation in the evaluated user-days, and five seeds give only a coarse estimate of that variation. The CERT web logs and the LANL authentication log were subsampled to make processing feasible. The sampling rate is identical for all users and independent of labels, but it reduces the evidence available per user-day, and a labelled user-day whose events are all removed cannot be observed (18 of the 176 LANL red-team user-days). The reported values therefore refer to the sampled event universe rather than to the complete logs. Finally, the temporal graph detector was evaluated without explicit time encoding in order to exclude target-derived timestamp statistics. Configurations with a source-only or causal time encoding were not evaluated.

VIII. Conclusion and Future Work

This study presents, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. The benchmark uses label-independent event sampling, a common user-day prediction unit, chronological and user-disjoint in-distribution controls, and an evaluable-set generalization gap, and no labelled or unlabelled target observations are used during training. Within-domain performance was heterogeneous, and transfer performance was substantially lower on average. For the Random Forest reference model, mean PR-AUC fell from 0.542 in distribution to 0.133 under zero-shot transfer, a generalization gap of 0.409. Transfer into SPEDIA was close to the target base rate, and transfer into and out of LANL remained at the performance floor. A memory-based temporal graph detector without explicit time encoding outperformed the Random Forest for CERT r4.2→SPEDIA but underperformed it for CERT r5.2→SPEDIA. The representation effect was therefore source-dependent, and no general graph advantage is claimed. Strong single-dataset performance is insufficient evidence of cross-dataset generalization.

Future work will extend the benchmark to further independently constructed corpora, including TWOS and DARPA OpTC, and will evaluate the temporal graph detector on the complete transfer matrix. It will also examine source-only or causal time encodings and target-assisted domain adaptation as separately reported settings, evaluate event-, session-, host-, and process-level prediction units, assess calibration transfer and operating-point stability, and test alternative action groupings. Future work will also examine whether attribution profiles remain stable across domains.

Declarations

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Data Availability

This study uses only third-party, publicly documented datasets, which are not redistributed by the authors. Each is available from its original source under its respective licence: the CERT Insider Threat Test Dataset; the SPEDIA dataset (Zenodo, https://doi.org/10.5281/zenodo.15495572); and the LANL Comprehensive Multi-Source Cyber-Security Events dataset.

Code Availability

The code (including the configuration of the camera-ready experiments), benchmark documentation and summary result tables supporting the camera-ready experiments are publicly available at https://github.com/alriyami7777-alt/crossdataset-insider-threat. The third-party datasets are not redistributed and must be obtained from their original sources.

Author Contributions

QASIM Mohamed Muhanna Alriyami: Conceptualization, Methodology, Software, Investigation, Formal analysis, Data curation, Validation, Visualization, Writing – original draft, Writing – review and editing. Mohd Murtadha Bin Mohamad: Supervision, Methodology, Validation, Writing – review and editing, Project administration.

Declaration of Generative AI Use

Generative AI tools were used during manuscript preparation for drafting assistance, language refinement and presentation support. All scientific decisions, experimental results, interpretations and final manuscript content were critically reviewed and verified by the authors, who take full responsibility for the work.

Ethical Approval

This work did not involve human participants or animals and did not collect any new personal data. All experiments were conducted on previously published, de-identified cybersecurity datasets; no ethical approval was therefore required.

REFERENCES

[1] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *Proc. IEEE Security and Privacy Workshops*, 2013, pp. 98–104. https://doi.org/10.1109/SPW.2013.37

[2] Y. Qi, C. Yan, Z. Wang, C. Zhang, S. Liu, Z. Lu, and B. Jiang, "ATHITD: Attention-based temporal heterogeneous graph neural network for insider threat detection," *Computers & Security*, vol. 157, p. 104587, 2025. https://doi.org/10.1016/j.cose.2025.104587

[3] K. Fei, J. Zhou, L. Su, W. Wang, and Y. Chen, "Log2Graph: A graph convolution neural network based method for insider threat detection," *Journal of Computer Security*, vol. 33, no. 1, pp. 37–56, 2025. https://doi.org/10.3233/JCS-230092

[4] X. Cai *et al.*, "LAN: Learning adaptive neighbors for real-time insider threat detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 10157–10172, 2024. https://doi.org/10.1109/TIFS.2024.3488527

[5] P. Pal, P. Chattopadhyay, and M. Swarnkar, "Temporal feature aggregation with attention for insider threat detection from activity logs," *Expert Systems with Applications*, vol. 224, p. 119925, 2023. https://doi.org/10.1016/j.eswa.2023.119925

[6] A. Alshehri, "Relational deep learning detection with multi-sequence representation for insider threats," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 758–765, 2022. https://doi.org/10.14569/IJACSA.2022.0130587

[7] K. Kundiya and Y. Haribhakta, "A systematic review on insider threat detection using natural language processing," *International Journal of Information Security*, vol. 24, no. 6, art. 227, 2025. https://doi.org/10.1007/s10207-025-01145-6

[8] S. Layeghy and M. Portmann, "On generalisability of machine learning-based network intrusion detection systems," *arXiv preprint*, 2022. https://doi.org/10.48550/arXiv.2205.04112

[9] M. Cantone, C. Marrocco, and A. Bria, "On the cross-dataset generalization of machine learning for network intrusion detection," *arXiv preprint*, 2024. https://doi.org/10.48550/arXiv.2402.10974

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," *Big Data Research*, vol. 30, p. 100359, 2022. https://doi.org/10.1016/j.bdr.2022.100359

[11] O. Oladejo and A. A. Ahmed, "Leveraging cross-domain transfer learning for enhanced multi-protocol network intrusion detection," *Computers*, vol. 15, no. 6, art. 376, 2026. https://doi.org/10.3390/computers15060376

[12] D. Álvarez, L. Pérez, A. Mateo, X. Larriva-Novo, M. Álvarez-Campana, and V. A. Villagrá, "SPEDIA dataset," *Zenodo*, 2025. https://doi.org/10.5281/zenodo.15495572

[13] A. D. Kent, "Comprehensive, multi-source cyber-security events data set," Los Alamos National Laboratory, 2015. https://doi.org/10.17021/1179829

[14] A. K. Paduvilan, S. A. Babu, J. Sekar, A. K. Elengovan, N. Seshagiri, K. Jangiti, and A. K. Selvam, "Attention-driven fusion of sequential and graph neural models for insider threat detection in UEBA," *Scientific Reports*, vol. 16, art. 21117, 2026. https://doi.org/10.1038/s41598-026-41242-7

[15] A. Harilal, F. Toffalini, J. Castellanos, J. Guarnizo, I. Homoliak, and M. Ochoa, "TWOS: A dataset of malicious insider threat behavior based on a gamified competition," in *Proc. MIST@CCS*, 2017, pp. 45–56. https://doi.org/10.1145/3139923.3139929

[16] D. C. Le and N. Zincir-Heywood, "Exploring anomalous behaviour detection and classification for insider threat identification," *Int. J. Network Management*, vol. 31, no. 4, p. e2109, 2021. https://doi.org/10.1002/nem.2109

[17] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020. https://doi.org/10.48550/arXiv.2006.10637

[18] D. Xu, C. Ruan, E. Körpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020. https://doi.org/10.48550/arXiv.2002.07962

Empirical Evaluation of Language Models for Arabic Financial Sentiment Analysis: A Comparative Study of Pretrained and General Large Language Models

Abdullah Almusned*, Mohammad Mehedi Hassan, Muhammad Al-Qurishi, Bader Alkhamees
Department of Information System-College of Computer and Information Sciences, King Saud University, Saudi Arabia

***Abstract*—This study presents a comparative study on the effectiveness of large and pre-trained language models in analyzing sentiment in Arabic financial news. While sentiment analysis using AI has advanced significantly in English-language finance, similar progress in Arabic remains limited. Most existing Arabic models, such as AraBERT and CAMeLBERT, are trained on general or social media posts and struggle with domain-specific content like financial data. At the same time, modern LLMs like ChatGPT and Gemini have not been thoroughly tested on Arabic financial data. Our work addresses these gaps by evaluating five models—ChatGPT, Gemini, Nuha (an Arabic LLM), AraBERT, and CAMeLBERT—on a dataset of 10,166 financial news headlines from Argaam, a leading Saudi financial platform. We conduct three key experiments. First, we conduct a standard sentiment classification comparison across all models. Second, prompt engineering experiment using zero-shot, one-shot, and three-shot strategies. Finally, persona-driven analysis where models respond from the viewpoint of different financial roles. The results show that Nuha consistently delivers the highest performance, except when prompts include user roles. They show promising results, but did not achieve the highest performance. ChatGPT maintains stable results across prompt variations and shows the most remarkable results when there are persona-driven prompts. Gemini performs well in zero-shot but degrades with added context. PLMs underperform, highlighting the need for better domain adaptation. Overall, this study provides practical insights into model behavior under different prompting conditions, emphasizes the importance of cultural alignment, and introduces persona-driven prompting as a valuable strategy for Arabic financial sentiment analysis.**

***Keywords*—Arabic inancial news; stock news; inancial news; domain speci ic; ChatGPT; Gemini; large language model; prompt strategies; Saudi stocks; sentiment analysis**

I. Introduction

The world of finance rapidly evolving thanks to technology like digital payments, online banking, and new ways to check identity and assess risk. At the same time, artificial intelligence (AI) makes understanding financial data easier. Machine learning (ML), and especially Natural Language Processing (NLP), now plays a significant role in analyzing stock markets [1], predicting trends [2], and spotting early signs of financial trouble [3]. One key task in this field is sentiment analysis, which looks at how people feel based on text found in places like financial news [4] [5], social media posts [6], or official company updates. These signals help investors make informed investment decisions and reduce risk by helping them understand how news might affect stock prices [7],[8].

In the past, researchers used simple machine learning tools and small datasets to study sentiment. Nevertheless, now, more powerful tools such as Pre-trained Language Models (PLMs) and Large Language Models (LLMs) make it possible to handle complicated language and context much better [9],[10]. For example, FinBERT was explicitly designed to understand English financial text and performs well on tasks that require understanding of market tone [11]. Some general-purpose models such as ChatGPT and Gemini have also been tested in different financial settings and languages, especially English, demonstrating competitive performance with proper input [12], [13].

However, despite these advances in English financial NLP, comparable empirical evidence for Arabic financial sentiment analysis remains limited—particularly in stock market news. Arabic models such as AraBERT [14] and CAMeLBERT [15] were mostly tested on general texts or social media posts, like telecom reviews [16] or tweets [17],[18]. These models face challenges due to Arabic's complex grammar, various dialects, and a lack of labeled data [19].

Our work helps fill that gap. It compares how well ChatGPT and Gemini (two powerful LLMs) perform against Nuha (an Arabic local LLM) and traditional PLMs like AraBERT and CAMeLBERT. The evaluation uses over 10,000 headlines from Argaam, a trusted Saudi stock market news source spanning multiple sectors. This research seeks to identify the sentiment of each news headline, leading to improved decision-making in the financial market. What makes it unique is not just the comparison of models but also two new ideas it tests. First, it looks at how zero-shot, one-shot, and three-shot prompting styles affect LLM model accuracy—a topic not widely explored in Arabic NLP. Second, it tries persona-based prompting by asking the LLM models to act like real-life users, such as financial analysts or traders, to see if that improves their answers. So far, to the best of our knowledge, this is one of the first studies to compare the performance of Arabic PLMs with modern LLMs on actual Arabic financial texts.

In view of the limited empirical evidence on Arabic financial sentiment analysis using modern language models, and the increasing relevance of prompt formulation and contextual framing, this study aims to systematically investigate how model architecture and prompting strategies affect sentiment

*Corresponding author.

classification performance in Saudi financial news. Accordingly, the following research questions are formulated:

- *RQ1:* How do instruction-tuned large language models compare with Arabic pretrained language models in classifying sentiment in Saudi financial news headlines?
- *RQ2:* How do different prompting strategies (zero-shot, one-shot, and few-shot) influence sentiment classification performance across large language models?
- *RQ3:* Does persona-driven prompting enhance the accuracy and robustness of sentiment classification in Arabic financial news analysis?

The rest of the study is as follows. Section II, is the conceptual and theoretical IV reports the results. Section V discusses the findings. Section VI will be the conclusion.

II. Conceptual and Theoretical Framework

This study is grounded in several interconnected theoretical foundations from natural language processing and machine learning that collectively shape the understanding of Arabic financial sentiment analysis. These foundations provide the conceptual perspective through which model behavior is interpreted and experimental methodologies are theoretically supported.

In-context learning, introduced by [20], refers to the capability of large language models to acquire task-specific behaviors directly from examples embedded within prompts, without requiring parameter updates or gradient-based optimization. This paradigm departs from conventional supervised learning, which depends on extensive fine-tuning using labeled datasets. Authors in [21] demonstrated that the effectiveness of in-context learning depends not merely on the presence of examples, but on their quality, relevance, and structural alignment with inference instances, indicating that models infer latent task structures rather than memorize input-output mappings. Authors in [22] further expanded this understanding through chain-of-thought prompting, showing that explicitly structured reasoning steps facilitate complex multi-stage inference. However, [23] reported substantial variability in few-shot performance across different model architectures and training objectives.

Transfer learning principles, formalized by [24] , posit that knowledge acquired during large-scale pre-training can be transferred to specialized downstream tasks, with effectiveness governed by domain proximity between source and target domains. Authors in [25] operationalized this concept through Universal Language Model Fine-tuning, demonstrating that sequential adaptation from general to domain-specific corpora outperforms task-specific training from scratch. Authors in [26] further showed that continued pre-training on domain-specific data yields significant performance gains when target domains exhibit specialized vocabulary and discourse patterns. In Arabic financial sentiment analysis, domain adaptation spans both linguistic dimensions—Modern Standard Arabic augmented with financial terminology—and topical dimensions associated with economic and market-related discourse. Comparisons between multilingual models (ChatGPT, Gemini), Arabic-local models (Nuha), and Arabic-specific pretrained models (AraBERT, CAMeLBERT) enable examination of whether linguistic and topical alignment corresponds with classification performance.

A fundamental distinction in contemporary language models exists between architectures trained solely on language modeling objectives and those subjected to instruction-tuning, which involves explicit optimization for following natural language instructions. Authors in [27] demonstrated through InstructGPT that fine-tuning models on diverse instruction-following tasks using reinforcement learning from human feedback substantially improves zero-shot generalization and task alignment. This approach is grounded in meta-learning principles, whereby exposure to large collections of instruction-formatted tasks enables models to infer task intent and adapt behavior accordingly [28],[29]. Authors in [30] further established that instruction-tuning yields cumulative benefits, improving both zero-shot and few-shot performance. This distinction differentiates instruction-tuned large language models from pretrained language models that rely primarily on masked language modeling objectives.

Prompt engineering has emerged as a central factor influencing large language model effectiveness, with systematic variations in input formulation producing substantial performance differences. Authors in [31] showed that minor changes in wording, phrasing, or structural organization can lead to notable accuracy fluctuations. Authors in [32] identified prompt sensitivity as a fundamental challenge, demonstrating that models may exhibit unstable performance across semantically equivalent but syntactically different prompts. Authors in [33] further documented extreme sensitivity to demonstration ordering in few-shot settings, showing that permuting identical examples can significantly affect outcomes.

Recent studies indicate that large language models can simulate distinct perspectives when explicitly assigned roles, with persona-based prompting influencing reasoning processes and task performance. Authors in [34] demonstrated that multi-persona approaches enhance complex problem-solving by activating different knowledge distributions within learned representations. Authors in [35] provided theoretical grounding, arguing that role-based prompting leverages models' exposure during pre-training to diverse professional registers and domain-specific discourse patterns. Authors in [36] further showed that persona prompts systematically shift model behavior and can improve performance when role framing aligns with domain expertise. In financial discourse, role-based prompting offers a structured means of contextualization, reflecting the differing analytical priorities of analysts, traders, and financial advisors.

Financial sentiment analysis represents a specialized domain characterized by distinctive linguistic properties and a strong dependence on numerical and temporal context. Authors in [37] demonstrated that general-purpose sentiment lexicons systematically misclassify financial texts because commonly used words often carry domain-specific meanings. Authors in [11] operationalized this insight through FinBERT, achieving state-of-the-art results by modeling finance-specific semantic patterns. Authors in [38] further established that financial sentiment is frequently expressed implicitly through factual reporting rather than overt opinion markers, requiring models to integrate numerical reasoning with linguistic interpretation.

In Arabic financial discourse, these challenges are compounded by language-specific factors, including morphological richness, formal registers, and culturally grounded indirect communication styles, which introduce additional layers of semantic complexity.

Multilingual language models aim to learn shared representations across languages, enabling cross-lingual transfer from high-resource to low-resource contexts. Authors in [39] demonstrated this capability through XLM-RoBERTa, achieving competitive multilingual performance despite substantial imbalance in training data. Authors in [40] found that multilingual models acquire language-agnostic syntactic and semantic abstractions. However, [41] reported that cross-lingual transfer degrades for morphologically rich languages such as Arabic, particularly in tasks requiring cultural grounding or domain-specific reasoning. Authors in [42] further showed that multilingual models can support few-shot cross-lingual transfer, though effectiveness depends on linguistic typology and representational quality. These findings highlight the complexities of multilingual modeling in specialized financial domains.

Based on the theoretical foundations outlined above, this study proposes the following hypotheses to guide the empirical investigation:

- H1: Instruction-tuned large language models will outperform Arabic pretrained language models in zero-shot Arabic financial sentiment classification, attributable to their enhanced instruction-following capabilities and higher-level contextual abstraction.
- H2: The impact of prompting strategies will be model-dependent, such that few-shot prompting will benefit models exhibiting strong in-context learning alignment, whereas models optimized primarily for abstraction-based generalization may experience performance degradation when exposed to demonstration-based prompts.
- H3: Persona-driven prompting will improve sentiment classification performance by introducing role-specific contextual framing that aligns model reasoning with domain-relevant financial decision perspectives.

III. Methodology

This section describes the different experiments used to evaluate the performance of large language models (LLMs) and pretrained language models (PLMs) in Arabic financial sentiment classification as it shown in Fig. 1. We start by defining the datasets and their sources. Then, we define each LLM and PLM model used in this study. Afterward, we introduce the different experiments. Finally, we set and describe the evaluation metrics.

A. Dataset

The dataset utilized in our work consisted of 10,166 Saudi stock market news headlines collected from Argaam [43], a leading financial economic website in the region. Argaam is owned by the Saudi Research and Media Group, a publicly listed company on the Saudi Stock Market. The headlines covered the period from January 1, 2021, to January 1, 2024, and covered 233 listed companies from 11 different sectors in the Saudi market. This comprehensive dataset offered a diverse representation of sentiments and writing styles, with the concise and focused nature of the headlines presenting both a valuable and challenging source for sentiment analysis due to their often implicit emotional small difference.

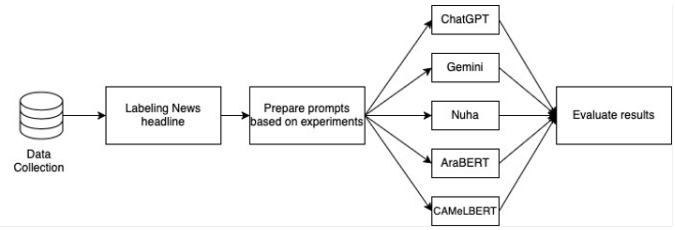


Fig. 1. Used methodology.

Each headline was assigned one of three sentiment labels: positive, negative, or neutral. The annotation process followed a hybrid approach, where ChatGPT, Gemini, and Nuha generated the initial labels, followed by manual validation by experts to determine the final reference labels.

B. Models

ChatGPT is a state-of-the-art large language model developed by OpenAI, built on the GPT (Generative Pretrained Transformer) architecture. It has been fine-tuned using reinforcement learning from human feedback (RLHF), which enhances its ability to generate contextually appropriate and coherent responses. Due to its extensive training on diverse and large-scale datasets, ChatGPT exhibits strong generalization capabilities, making it well-suited for tasks such as sentiment analysis, especially in domains like financial news where contextual nuance and linguistic variability are critical. In this study, we used GPT via APIs with a temperature of 0.1.

Gemini is Google DeepMind's advanced large language model, designed to compete at the frontier of generative AI. Leveraging cutting-edge transformer architectures and trained on multilingual, multi-modal datasets, Gemini aims to deliver high performance across a range of tasks, including natural language understanding and generation. Its multilingual capabilities and powerful reasoning abilities make it a promising candidate for complex applications such as sentiment classification in finance, particularly when dealing with rich, structured data inputs. For the purpose of this study, all safety settings were disabled to ensure that the model's raw interpretive capacity could be evaluated without filtering or moderation constraints.

Nuha, previously referred to as Juhaina, is a bilingual Arabic-English Large Language Model developed by Elm Company with the explicit goal of cultural and linguistic alignment with Arabic-speaking users. Nuha supports a wide range of tasks including instruction-following, open-domain question answering, and sentiment classification. It is post-trained using supervised fine-tuning (SFT) and human preference alignment methods to improve its generation quality and adherence to Arabic cultural norms. In our work, Nuha is evaluated alongside other LLMs to assess its performance on Arabic financial sentiment analysis, contributing to the exploration of locally aligned generative AI solutions.

CAMelBERT is a family of Arabic pretrained language models introduced by the CAMeL Lab at New York University Abu Dhabi. The model variants are trained on extensive corpora representing Modern Standard Arabic (MSA), Dialectal Arabic (DA), Classical Arabic (CA), and a mixture of these variants. CAMeLBERT adopts the BERT architecture with standardized pretraining procedures, leveraging over 167 GB of high-quality Arabic text. Through systematic experimentation, the developers demonstrated that language variant proximity—rather than data volume—is the most significant factor influencing downstream task performance. In this study, CAMeLBERT is used as an Arabic PLM, offering valuable insight into how diverse pretraining data affects sentiment analysis outcomes. Therefore, the 'bert-base-arabic-camelbert-mix-sentiment' model was employed without additional fine-tuning.

AraBERT is a monolingual Arabic language model based on the BERT-base architecture, developed by researchers at the American University of Beirut. It consists of 12 transformer layers and approximately 110 million parameters. AraBERT was trained using the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives on a large Arabic corpus comprising 70 million sentences, collected from Arabic Wikipedia, the 1.5B words El-Khair corpus, and the OSIAN news corpus. AraBERT has shown state-of-the-art results across various Arabic NLP tasks such as sentiment analysis, named entity recognition, and question answering. In our study, AraBERT serves as an Arabic Pretrained Language Model (PLM) for sentiment classification. Accordingly, the 'AraBert-Arabic-Sentiment-Analysis' model was utilized in its pretrained configuration, without fine-tuning.

C. Experiments

We conducted a series of experiments using multiple prompting strategies across general-purpose and Arabic-specific language models on Arabic financial sentiment data. This section defines each experiment, and the prompt used for it.

1) Experiment 1: Comparative analysis of sentiment analysis models: The first experiment compared five sentiment analysis models—AraBERT, CAMeLBERT, ChatGPT, Gemini, and Nuha—on their ability to classify Arabic financial news into three sentiment categories: positive, negative, and neutral. Each model was applied to the same dataset of headlines, ensuring a consistent baseline for comparison as shown in Table I (Exp1). AraBERT and CAMeLBERT were leveraging their pre-training on Arabic text. ChatGPT, Gemini and Nuha, as generative language models, were evaluated in zero-shot settings. This experiment aimed to provide insights into the models' performance across a standard sentiment classification task.

2) Experiment 2: Zero-shot, one-shot, and few-shot prompt strategies: The second experiment focused on comparing the performance of ChatGPT, Gemini, and Nuha under different prompting approaches: zero-shot, one-shot, and few-shot strategies. In this setup, the models were tasked with sentiment classification of Arabic news headlines. Zero-shot prompting involved no prior examples, one-shot used a single example within the prompt, and few-shot included three examples of each class to guide the models as shown in Table I (Exp2). This experiment was designed to evaluate how the inclusion of examples influenced the models' ability to accurately interpret and classify sentiments, providing deeper insights into their adaptability and learning capabilities. CAMeLBERT and AraBERT were not evaluated due to their limited capabilities in zero/few-shot contexts and absence of instruction-tuned architectures, but given their baseline performance in Experiment 1, their inclusion would likely have yielded similarly weak results under complex prompting scenarios.

TABLE I. PERSONA-DRIVEN PROMPTS

ID	Prompt
Exp1	Respond with only one word (*neutral*, *positive*, or *negative*) based on the sentiment of the input text.
P1	Act as a financial analyst expert in Saudi stocks. How would you assess the tone of the news in the following headline in relation to the stock? Respond only in one word: *positive*, *negative*, or *neutral*. Headline: {headline}.
P2	Act as a Saudi stock expert trader. Based on the headline {headline}, how would you interpret the news regarding its potential impact on the stock? Respond only in one word: *positive*, *negative*, or *neutral*.
P3	Act as a sentiment analysis model trained on Saudi financial news headlines. Based on the headline {headline}, Respond only in one word: *positive*, *negative*, or *neutral*.
P4	Act as an expert in Saudi stock trading. Based only on the headline, will you buy, sell, or hold your stocks the next day? Respond only in one word: *positive*, *negative*, or *neutral* (positive = buy, negative = sell, neutral = hold). Headline: {headline}.

Prompts used to evaluate the effect of different expert personas on sentiment classification of Saudi stock market news headlines.

3) Experiment 3: Persona-driven prompts based on user roles: The third experiment explored the use of role-specific personalized prompts to adapt sentiment analysis outputs based on the perspective of specific professional roles, such as financial analysts, Saudi traders, and other relevant job profiles. For this purpose, ChatGPT, Gemini, and Nuha were tested using tailored prompts designed to align the sentiment classification process with the priorities and contextual needs of these roles. Table I (P1-P4) details the prompts used for sentiment classification. For example, prompts for financial analysts emphasized sentiment relevant to market trends and economic impact, while prompts for Saudi traders focused on actionable insights and immediate relevance to trading activities. This experiment aimed to evaluate the potential of personalized prompts in enhancing the contextual appropriateness and practical utility of sentiment analysis. For the same reason in the second experiment, CAMeLBERT and AraBERT were not evaluated.

D. Evaluation Metrics

To evaluate the performance of the five models in sentiment classification task, we employed a standard suite of metrics. These encompassed accuracy, precision, recall, F1-score, Cohen's Kappa, and confusion matrices.

Accuracy measures the proportion of correctly classified instances. However, it can be misleading with imbalanced datasets.

Precision, Recall, and F1-score were used to address accuracy's limitations, we used precision (true positives among

predicted positives), recall (true positives among actual positives), and their harmonic mean, the F1-score. These were calculated per class and using macro- and weighted-averaging to account for class imbalance.

Cohen's Kappa measures agreement between predicted and true labels, correcting for chance agreement. A Kappa of 1 indicates perfect agreement, 0 indicates chance agreement, and -1 indicates perfect disagreement. This is particularly useful for comparing models with different class biases.

The Confusion Matrix provides a tabular visualization of classification performance, showing true/false positives/negatives for each class and aiding in identifying misclassifications patterns.

IV. Results

This section presents the findings intended as descriptive comparisons from three experiments designed to evaluate the performance of five large language models, ChatGPT, Gemini, Nuha, CAMeLBERT, and AraBERT, in the sentiment classification tasks. The models were assessed using standard performance metrics, accuracy, precision, recall, F1-score, and Cohen's Kappa. The following subsections detail the outcomes of each experiment, highlighting key patterns and comparative insights.

A. Experiment 1: Comparative Analysis of Sentiment Analysis Models

The evaluation results presented in the comparative Table II offer significant insights into the performance of five language models: ChatGPT, Gemini, Nuha, CAMeLBERT, and AraBERT. Among the evaluated models, Nuha achieved the highest observed performance, with an accuracy of 89.09%, a precision of 0.8958, a recall of 0.8909, and an F1-score of 0.8914. This suggests Nuha is highly effective in distinguishing between different sentiment categories while maintaining a strong balance between precision and recall. Gemini follows closely, with an accuracy of 88.69% and the highest precision 0.9027, indicating a conservative classification strategy that prioritizes correctness in positive sentiment predictions. However, its recall is marginally lower than Nuha. ChatGPT, while slightly trailing behind Gemini and Nuha, still performs competitively with an accuracy of 86.70%, a precision of 0.8861, a recall of 0.8670, and an F1-score of 0.8633, demonstrating a strong generalization capability.

Conversely, CAMeLBERT and AraBERT exhibit significantly weaker performance, with CAMeLBERT achieving 60.21% accuracy and AraBERT scoring the lowest at 55.14%. These results indicate their limited ability to accurately classify Arabic sentiment, likely due to suboptimal domain adaptation. The confusion matrices further illustrate classification errors and model biases, shedding light on misclassification tendencies.

Nuha minimizes classification errors, correctly predicting 4,779 positive, 937 neutral, and 3,341 negative instances, with only misclassified 767 positives, 96 neutral and 246 negatives, reinforcing its robustness. Gemini, while strong in both precision and recall, struggles with false negatives, misclassifying 1,000 positive instances, which may indicate a conservative bias in sentiment classification. ChatGPT performs well in classifying 5,469 positive, 816 neutral and 2,529 negative sentiments correctly, but exhibits a relatively high number of false negatives of 1,058 misclassified negative instances, suggesting a tendency to overestimate negativity in certain contexts. Fig. 2 shows the confusion matrix for each model.

TABLE II. Experiment 1 Performance Metrics

Model	Accuracy (%)	Precision	Recall	F1-score
ChatGPT	86.70	0.8861	0.8670	0.8633
Gemini	88.69	0.9027	0.8869	0.8876
Nuha	**89.09**	**0.8958**	**0.8909**	**0.8914**
CAMeLBERT	60.21	0.7198	0.6012	0.5056
AraBERT	55.14	0.7325	0.5514	0.3989

Performance comparison of large language models and transformer-based baselines on Experiment 1. The best result in each metric is shown in bold.

Pre-trained models, CAMeLBERT and AraBERT exhibited significant classification confusion, particularly in differentiating neutral and positive sentiment. CAMeLBERT misclassified 956 neutral instances as positive, while AraBERT exhibits a strong bias toward predicting positive sentiment, incorrectly classifying 1,031 neutral instances as positive and 3,525 negative instances as positive. This suggests that AraBERT struggles to differentiate between neutral and positive sentiment polarities, a significant limitation for applications requiring precise sentiment analysis. Additionally, CAMeLBERT's high misclassification rate for negative sentiment where 576 instances are incorrectly labeled as positive, suggests it lacks strong contextual understanding of negative sentiment indicators.

From the confusion matrices, two key patterns emerge. First, bias toward positive sentiment is apparent in AraBERT and CAMeLBERT, with a disproportionate number of neutral and negative instances classified as positive. This suggests these models fail to capture sentiment variations, particularly in distinguishing neutral language from positive sentiment. Second, ChatGPT and Gemini exhibit higher false negative rates, where positive instances are sometimes misclassified as negative. While this could indicate a conservative classification strategy aimed at reducing false positives, it may lead to missed positive sentiment detections, which is crucial in sentiment-sensitive applications like our situation financial news.

To further assess inter-rater reliability and classification agreement, Cohen's Kappa coefficient was calculated for each model as it shows in Fig. 3. Cohen's Kappa is a more robust measure than accuracy as it accounts for chance agreement. The results reveal a clear distinction in model reliability. Nuha achieved 0.8107 and Gemini 0.8064 exhibited the highest agreement with ground truth, reinforcing their strong consistency in classification. ChatGPT with score 0.7537 follows closely, indicating good agreement but slightly more variance in predictions compared to Nuha and Gemini. CAMeLBERT achieved 0.1526 demonstrates poor agreement, aligning with its high misclassification rates seen in the confusion matrices. AraBERT with score 0.0159 exhibits an almost random agreement level, confirming its severe misclassification issues and bias toward predicting positive sentiment.

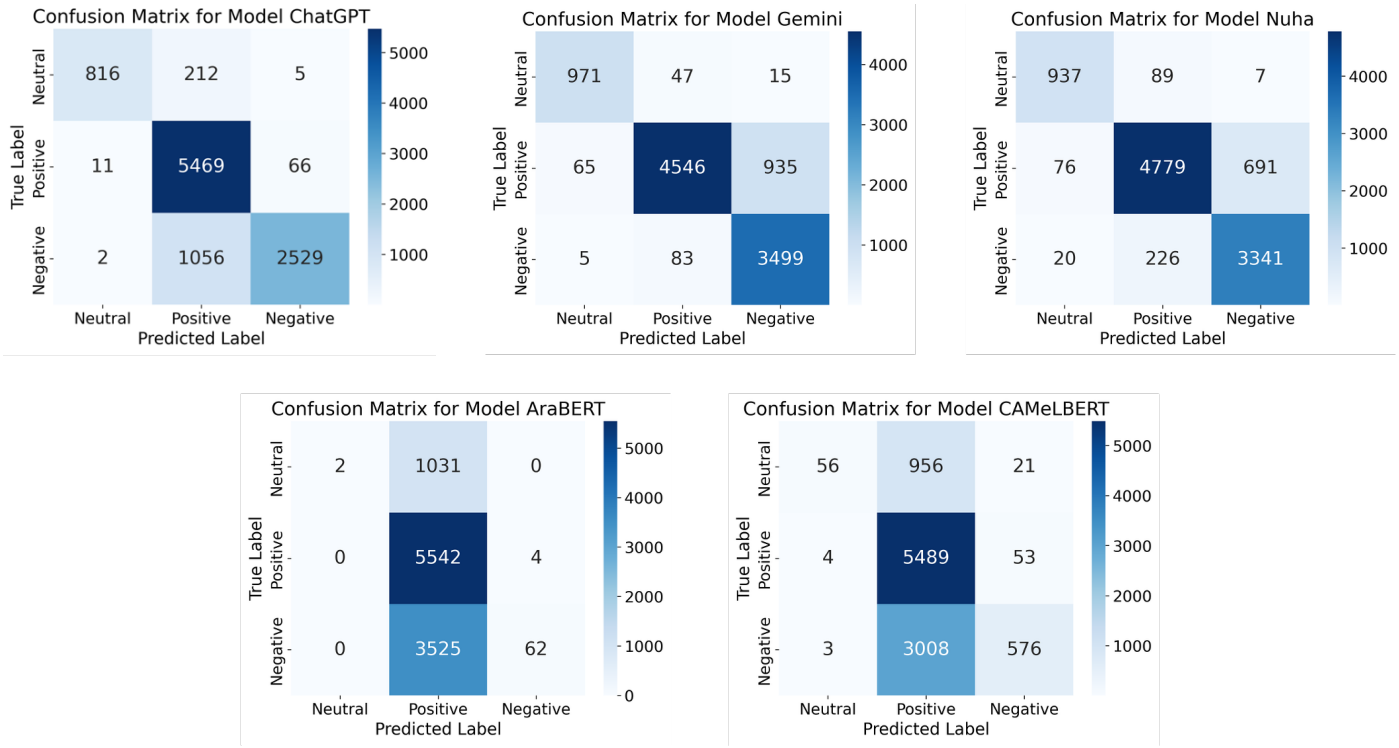


Fig. 2. Experiment 1 confusion matrix.

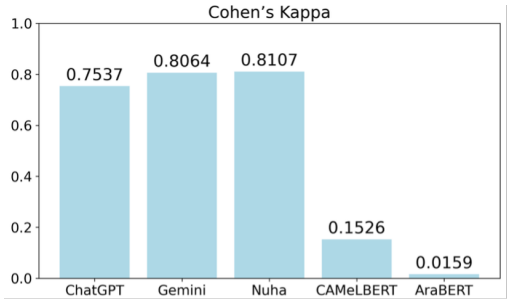


Fig. 3. Experiment 1 Cohen's kappa.

Overall, Nuha proves to be the most reliable model, excelling across all key performance metrics while maintaining low error rates across sentiment categories. Gemini and ChatGPT remain strong alternatives, with Gemini excelling in precision and ChatGPT offering a balanced classification approach. CAMeLBERT and AraBERT, however, require significant improvements, particularly in distinguishing between neutral and positive sentiment, as well as in reducing their overall misclassification rates. These findings reinforce the importance of careful model selection in sentiment analysis tasks, particularly in fields where classification precision directly impacts decision-making processes.

B. Experiment 2: Zero-Shot, One-Shot, and Three-Shot Prompt Strategies

The second experiment evaluates the impact of prompting strategy—specifically Zero-Shot, One-Shot, and Three-Shots—on the performance of three large language models: ChatGPT, Gemini, and Nuha. Table III summarized the performance metrics for this experiment.

TABLE III. EXPERIMENT 2 PERFORMANCE METRICS

Model	Strategy	Accuracy (%)	Precision	Recall	F1-score
ChatGPT	Zero	86.70	0.8861	0.8670	0.8633
	One	**87.40**	**0.8868**	**0.8740**	**0.8707**
	Three	83.43	0.8599	0.8343	0.8254
Gemini	Zero	**88.69**	**0.9027**	**0.8869**	**0.8876**
	One	58.34	0.5671	0.5834	0.5632
	Three	56.36	0.5439	0.5636	0.5256
Nuha	Zero	89.09	0.8958	0.8909	0.8914
	One	91.97	0.9200	0.9197	0.9197
	Three	**92.45**	**0.9247**	**0.9245**	**0.9245**

Performance comparison of different prompting strategies (zero-, one-, and three-shot) across the evaluated models in Experiment 2. The best result for each model is shown in bold.

Nuha emerged as the most consistently performant model across all prompt configurations. It achieved an accuracy of 89.09% in the zero-shot, improving to 91.97% in one-shot and further to 92.45% in the three-shot. Precision, recall, and F1-score followed the same ascending pattern. Cohen's Kappa values for Nuha also reflect strong inter-rater agreement, increasing steadily from 0.8107 to 0.8674. The confusion matrices [Fig. 4(c)] confirm this consistency, with Nuha classifying the majority of positive and negative instances correctly while maintaining balanced performance across all sentiment classes.

These consistent improvements suggest that Nuha benefits significantly from in-context learning, effectively incorporating example-driven prompts to enhance classification quality without suffering from overfitting or degradation.

ChatGPT demonstrated relatively stable performance across all prompting strategies, with minor fluctuations. Its zero-shot accuracy was 86.70%, marginally improving to 87.40% in one-shot, but decreasing to 83.43% in three-shots. Precision and recall peaked in the one-shot setting at 0.8868 and 0.8740, respectively, before declining in three-shots 0.8599 and 0.8343. The F1-score mirrored this trend, with values of 0.8633, 0.8707, and 0.8254. This slight decline in the three-shot setting suggests ChatGPT may struggle to integrate larger contextual prompts effectively. The Cohen's Kappa values of 0.7537, 0.7700, and 0.6931, respectively, support this interpretation, indicating strong agreement in zero- and one-shot, but reduced reliability when exposed to extended prompt length. The confusion matrices [Fig. 4(a)] further illustrate this trend that ChatGPT performs best with minimal context (zero- or one-shot), but misclassifies more negative instances as positive in three-shot, suggesting some semantic drift or context overload. Nevertheless, its relatively high and balanced scores across all metrics indicate that ChatGPT maintains a robust baseline performance and is generally resilient to prompt variation.

In contrast, Gemini exhibits a sharp and unexpected decline in performance when shifting from zero-shot to few-shot strategies. It achieves 88.69% accuracy, 0.8869 recall, and an F1-score of 0.8876 in zero-shot, performing comparably to Nuha and exceeding ChatGPT. However, in one-shot and three-shots, Gemini's accuracy drops to 58.34% and 56.36%, recall to 0.5834 and 0.5636, and F1-score to 0.5632 and 0.5256, respectively. The Cohen Kappa values also plummet 0.2185 and 0.1616, showing low agreement and near-random performance in those settings (Fig. 5). This pattern indicates that Gemini is highly sensitive to few-shot prompting, possibly due to inefficient handling of in-context learning examples or overfitting to provide demonstrations. the conclusion that Gemini excels in generalization without explicit examples, but fails to adapt when such examples are introduced.

The overall findings underscore three key insights. First, in-context prompting improves model performance only when the model architecture and training support such adaptation, as evidenced by Nuha's consistent gains. Second, not all models benefit from increasing the number of examples, with Gemini's performance declining sharply under few-shot conditions, suggesting possible overfitting or distraction effects. Third, ChatGPT offers balanced generalization, performing reliably across varying prompt styles but showing diminishing returns in more complex three-shot contexts.

In conclusion, the experiment highlights the non-linear and model-specific effects of prompting strategies. While Nuha leverages contextual prompts to enhance performance, and ChatGPT maintains stability, Gemini's decline under few-shot settings suggests structural or representational limitations that hinder its ability to utilize prompt examples. These findings emphasize the importance of understanding model behavior in relation to task design and prompt structure for optimal deployment of language models in sentiment analysis.

TABLE IV. EXPERIMENT 3 PERFORMANCE METRICS

Model	Persona	Accuracy (%)	Precision	Recall	F1-score
ChatGPT	P1	91.64	0.9208	0.9164	0.9166
	P2	89.41	0.9042	0.8941	0.8943
	P3	**92.84**	**0.9287**	**0.9284**	**0.9282**
	P4	90.66	0.9091	0.9066	0.9068
Gemini	P1	87.13	0.8940	0.8713	0.8719
	P2	85.92	0.8871	0.8592	0.8600
	P3	87.13	**0.8981**	0.8713	0.8718
	P4	**87.18**	0.8730	**0.8718**	**0.8721**
Nuha	P1	**87.66**	**0.8925**	**0.8766**	**0.8771**
	P2	85.72	0.8822	0.8572	0.8579
	P3	85.58	0.8776	0.8558	0.8564
	P4	86.68	0.8681	0.8668	0.8671

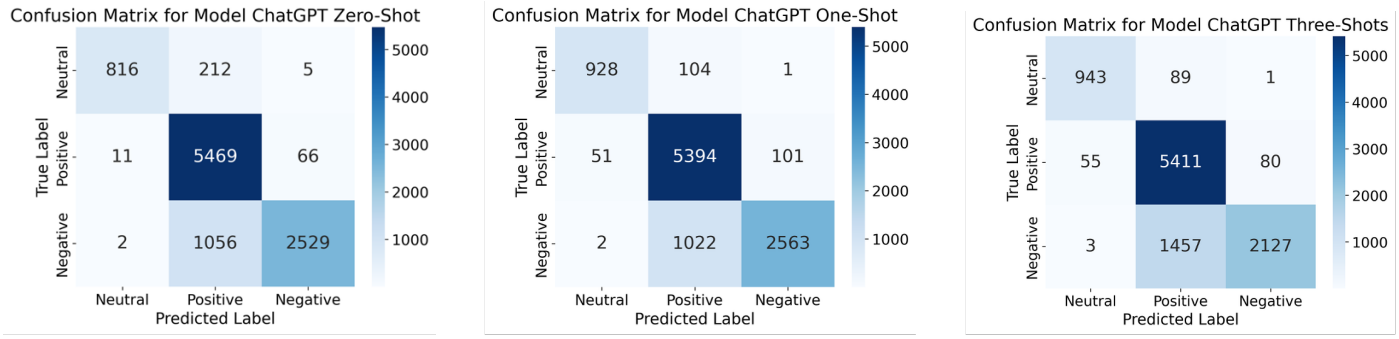
Performance comparison of the four persona-driven prompting strategies across the evaluated models in Experiment 3. The best result for each model is shown in bold.

C. Experiment 3: Persona-Driven Prompts Based on User Roles

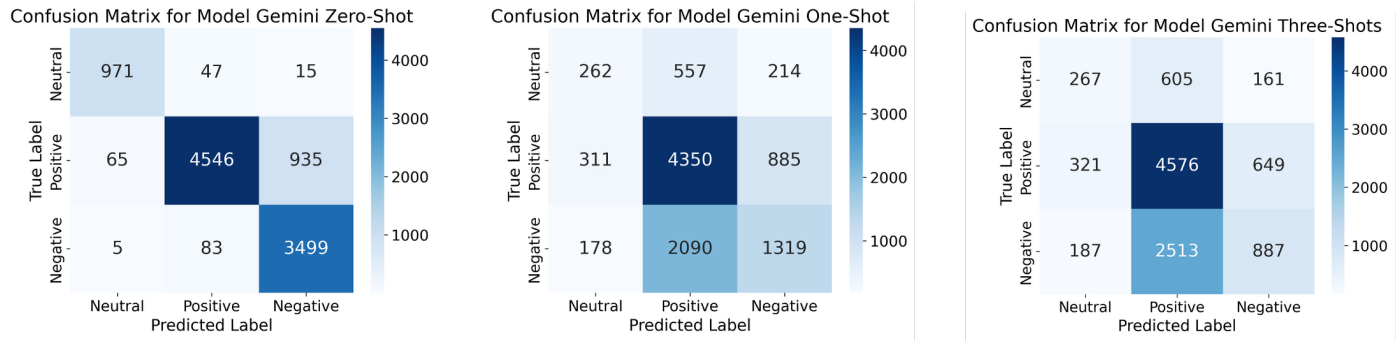
The third experiment explores the influence of persona-driven prompt variation (P1–P4) on the performance of three large language models (ChatGPT, Gemini, and Nuha) in sentiment analysis. Each persona represents a distinct style of prompting designed to assess how prompt framing affects classification quality, financial analyst, Saudi stock expert trader, sentiment analysis, and expert in Saudi stock recommendations, respectively. The models were evaluated using accuracy, precision, recall, F1-score, Cohen's Kappa, and supported by detailed confusion matrix analysis to assess class-specific behavior as is shown in Table IV, Fig. 6, and Fig. 7.

ChatGPT demonstrated the highest and most consistent performance across the four evaluated prompts. Its best performance was under P3 (sentiment analysis), where it achieved 92.84% accuracy, 0.9287 precision, 0.9284 recall, and 0.9282 F1-score, with a Cohen's Kappa of 0.8732. The confusion matrix for P3 reveals that ChatGPT correctly classified 5,271 positive, 971 neutral and 3,196 negative instances. Errors were mainly concentrated in misclassifying negative as positive in 385 cases, which is expected in closely aligned sentiment boundaries. Even under its lowest-performing persona, P2 (Saudi stock expert trader), ChatGPT retained 89.41% accuracy, and a solid balance of 0.9042 precision, 0.8941 recall, and 0.8943 F1-score. The P2 confusion matrix shows stronger neutral classification with 987 correct cases but slightly more misclassification of positive and negative examples compared to P3 [Fig. 6(a)]. Under P1 (Financial analyst) and P4 (expert in Saudi stock recommendations), ChatGPT continued to perform robustly, with Cohen's Kappa values of 0.8554 and 0.8380, respectively, and confusion matrices indicating high true positives and effective neutral distinction. These results collectively indicate that ChatGPT is highly responsive to persona framing, with P3 offering the best contextual synergy.

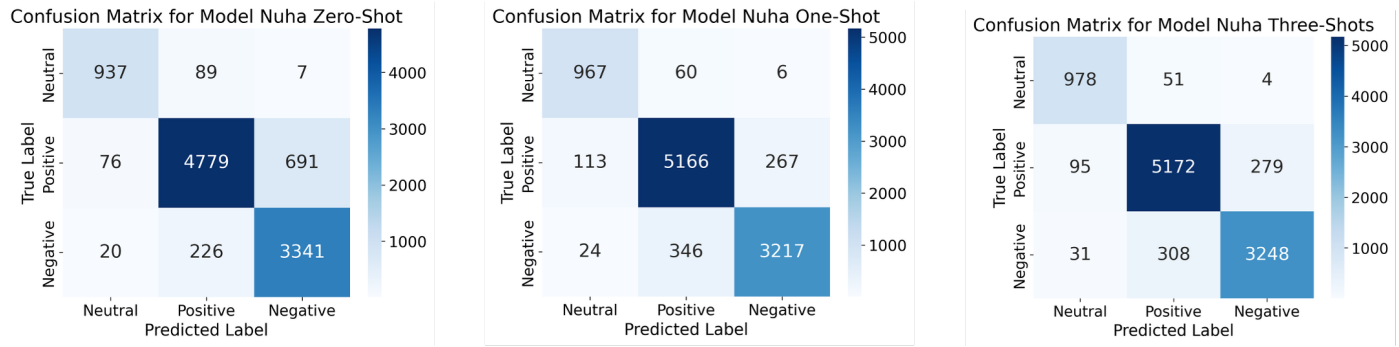
Gemini, while not reaching ChatGPT's peaks, showed strong and consistent performance across personas, with accuracy values ranging from 85.92% to 87.18%. Its most effective performance appeared in P1 (financial analyst) and P3 (sentiment analysis) both with approximately 87.13% accuracy, with



(a) ChatGPT



(b) Gemini



(c) Nuha

Fig. 4. Experiment 2 confusion matrix.

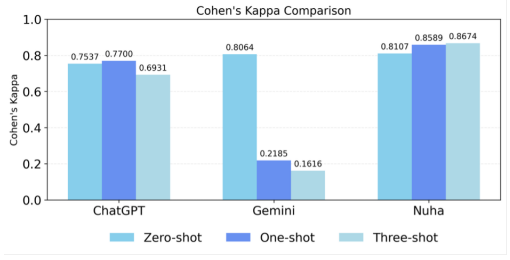
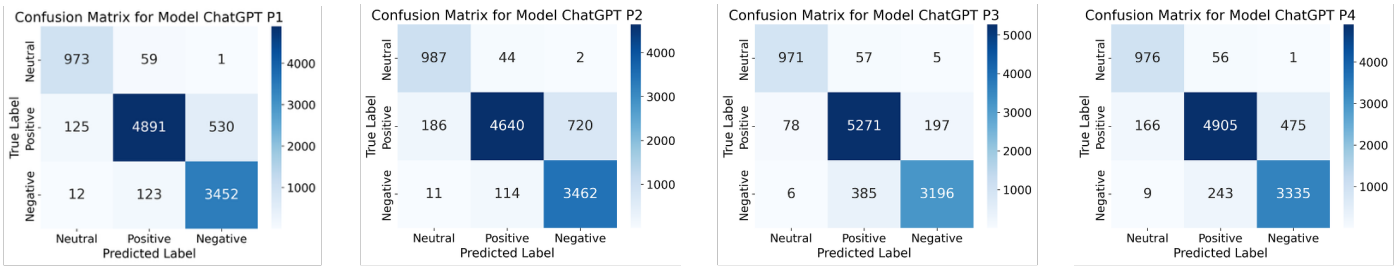


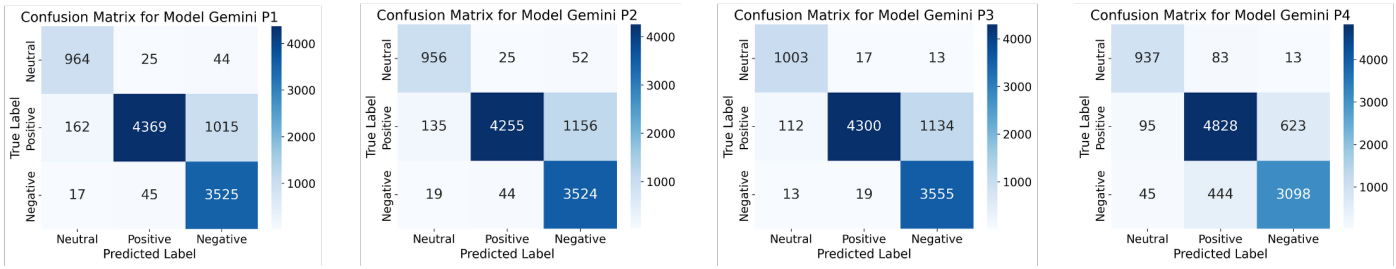
Fig. 5. Experiment 2 Cohen's kappa.

F1-scores of 0.8719 and 0.8718, and Cohen's Kappa values of 0.7824 and 0.7830, respectively. The confusion matrix for P1 indicates solid classification of negative sentiment with 3,525 correct cases and a modest level of misclassification between positive and negative 1,015 false-negatives. Under P3, Gemini improved neutral accuracy with 1,003 correct cases and slightly reduced false positives for positive sentiment [Fig. 6(b)]. P2 (Saudi stock expert trader) showed a drop in performance, with 1,208 false negatives and lower agreement 0.7625 Kappa, indicating sensitivity to specific persona structure. Despite this, Gemini's stability across personas reflects a moderate adaptability, suitable for prompt settings that are consistent and well-defined.

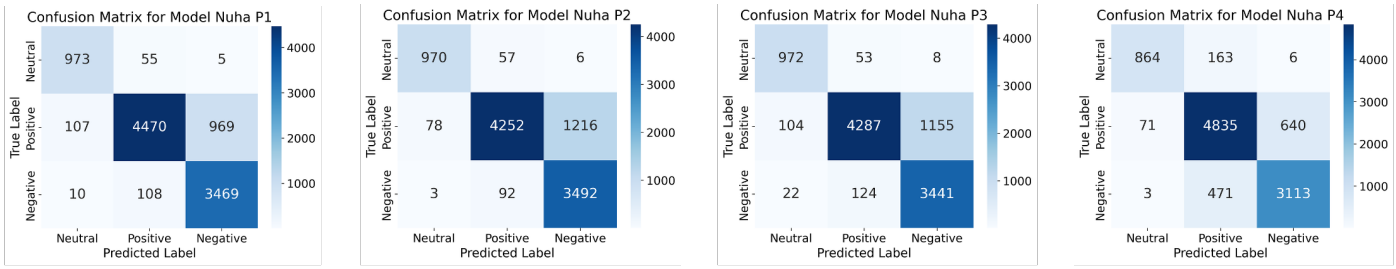
Nuha exhibited the greatest variance across personas, with accuracy ranging from 85.58% with P3 (sentiment analysis) to 87.66% with P1 (financial analyst). Precision, recall, and F1-score followed similar fluctuations. The best performance under P1 produced 0.8925 precision, 0.8766 recall, and 0.8771



(a) ChatGPT



(b) Gemini



(c) Nuha

Fig. 6. Experiment 3 confusion matrix.

F1-score, with a Cohen's Kappa of 0.7896. Nuha's P1 confusion matrix shows strong classification of positive 4,470 correct cases, neutral 973 correct cases and negative 3,469 correct cases examples but a notable number of identifiying 969 positives as negative cases. In P3, performance slightly declined: fewer positive and negative instances were correctly identified, with an increase in 1,155 false-positives cases and misclassified neutrals. P2 (Saudi stock expert trader) and P4 (expert in Saudi stock recommendations) followed a similar trend, with the P4 confusion matrix revealing the highest misclassification of neutral instances with 163 neutral examples classified as positive, which likely impacted overall precision [Fig. 6(c)]. Cohen's Kappa (Fig. 7) values for Nuha range from 0.7554 to 0.7896, which confirm its inconsistent agreement, particularly in neutral sentiment recognition, which may be influenced by prompt nuances embedded in each persona.

From an error analysis perspective, ChatGPT consistently showed the lowest number of high-impact errors—especially in distinguishing positive from negative sentiments—across all personas. Its three-way balance among neutral, positive, and negative classes was evident across confusion matrices, with P3 performing best. Gemini, while reliable in handling positive and negative sentiment, had greater difficulty distinguishing

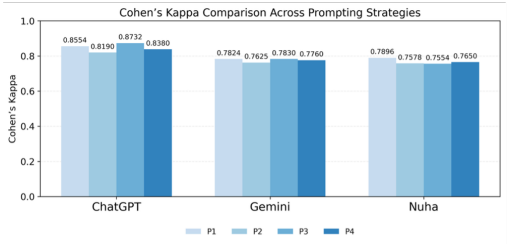


Fig. 7. Experiment 3 Cohen's kappa.

neutral sentiment, particularly under P2 and P4. Nuha struggled most in neutral sentiment detection, frequently confusing neutral with positive, especially in P4, where 163 neutral instances were misclassified as positive, lowering precision and Kappa agreement.

In terms of inter-rater reliability, ChatGPT outperformed both Gemini and Nuha, with Kappa values ranging from 0.8190 to 0.8732, showing strong and consistent agreement with ground truth across all personas. Gemini maintained moderate reliability from 0.7625 to 0.7830, while Nuha's reliability was more affected by personas variation between

0.7554 and 0.7896.

In summary, ChatGPT showed greater adaptability to persona-driven prompting, with the P3 configuration yielding the highest observed performance. Gemini remained stable and moderately effective, while Nuha exhibited a noticeable sensitivity to prompt framing, particularly in handling neutral sentiment. These findings suggest that prompt design—especially persona stylization—can significantly influence model behavior, and that ChatGPT is best suited for stock news. Consequently, persona tuning should be considered a critical component in applied sentiment analysis pipelines, especially when deploying large language models in real-world, user-facing environments, and domain-specific.

V. Discussion

This study investigated the performance of five large language models (ChatGPT, Gemini, Nuha, CAMeLBERT, and AraBERT) across three distinct experimental designs in the context of sentiment classification for the Saudi stock market financial news. Each experiment was developed to examine the models' capabilities from a different perspective: a direct zero-shot prompt (Experiment 1), prompt-based contextual learning using different shots techniques (Experiment 2), and persona-driven prompts based on user roles (Experiment 3). The evaluation utilized standard classification metrics—accuracy, precision, recall, and F1-score—supplemented by confusion matrix and Cohen's Kappa to measure inter-rater agreement beyond chance. Together, these experiments reveal critical insights into each model's behavior under varying prompting conditions.

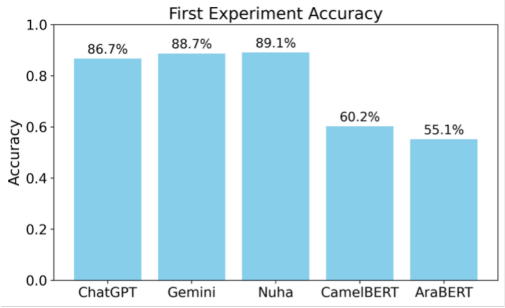


Fig. 8. Experiment 1 accuracy.

In the first experiment, which tested all five models on a standard classification task without prompt variation, Nuha, Gemini, and ChatGPT demonstrated high levels of performance as shown in Fig. 8, achieving accuracies of 89.09%, 88.69%, and 86.70%, respectively. These models also scored highly across precision, recall, and F1-score, suggesting strong overall balance. CAMeLBERT and AraBERT, in contrast, exhibited markedly lower performance. CAMeLBERT recorded an accuracy of 60.21%, and AraBERT dropped further to 55.14%, with corresponding F1-scores of 0.5056 and 0.3989. The significantly lower Kappa values for CAMeLBERT with 0.1526 and AraBERT with 0.0159 indicate poor agreement with ground truth labels, suggesting that these models lack the capacity for nuanced sentiment differentiation, particularly when operating without prompt customization or instruction.

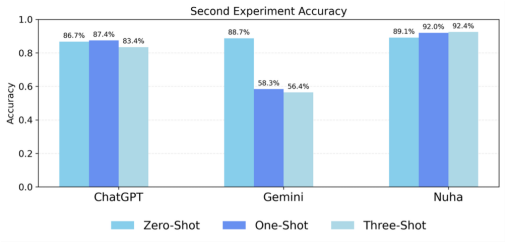


Fig. 9. Experiment 2 accuracy.

Regarding RQ1, the results demonstrate a pronounced performance gap between instruction-tuned large language models and Arabic pretrained language models in the task of Saudi financial sentiment classification. Across all evaluation metrics, ChatGPT, Gemini, and Nuha consistently outperformed AraBERT and CAMeLBERT under zero-shot conditions. This outcome supports the premise that instruction-tuned models benefit from exposure to diverse task formulations during training, enabling them to interpret sentiment analysis as an instruction-following problem rather than a fixed classification task. In contrast, AraBERT and CAMeLBERT—trained primarily through masked language modeling and dependent on downstream fine-tuning—showed limited adaptability when applied directly to financial sentiment classification without domain-specific retraining. Furthermore, the results indicate that the formal and implicitly evaluative nature of Saudi financial headlines, where sentiment is often embedded within regulatory and factual language, imposes high contextual reasoning demands. Such characteristics favor models capable of instruction-level abstraction over those relying predominantly on token-level pattern learning.

While in the second experiment, it introduced prompt-based context through zero-shot, one-shot, and three-shot prompting strategies. CAMeLBERT and AraBERT were not evaluated due to their absence of instruction-tuned architectures. Here, Nuha exhibited the most robust and consistent improvement (Fig. 9), culminating in a three-shot accuracy of 92.45%, and achieving the highest F1-score with 0.9245 and 0.8674 Kappa among all models. ChatGPT showed moderate gains in the one-shot condition with 87.40% accuracy but experienced a decline in the three-shot configuration at 83.43%, possibly due to increased prompt complexity or context saturation. Gemini, in contrast, experienced substantial degradation in performance as prompting complexity increased. While its zero-shot performance remained strong with 88.69% accuracy, its one-shot and three-shot accuracies fell to 58.34% and 56.36%, respectively, accompanied by significant drops in precision, recall, and Kappa scores.

With respect to RQ2, the experiments reveal that the effect of prompting strategies is highly model-dependent. Nuha consistently benefited from richer contextual input, with performance improving from zero-shot to one-shot and three-shot prompting, reflecting effective in-context learning and stable integration of demonstrations. ChatGPT exhibited moderate sensitivity, achieving optimal performance under one-shot prompting while showing a slight decline as prompt complexity increased, suggesting partial context saturation. In contrast, Gemini displayed a marked and unexpected perfor-

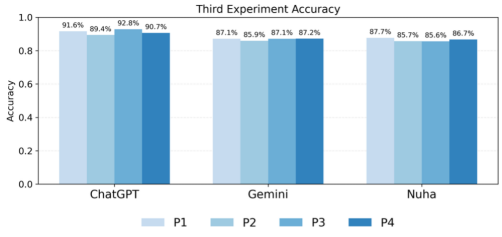


Fig. 10. Experiment 3 Accuracy.

mance degradation when transitioning from zero-shot to few-shot prompting. This behavior can be interpreted in terms of prompt interference, where models optimized for abstraction-based zero-shot generalization may experience misalignment when explicit demonstrations are introduced. These findings indicate that few-shot prompting does not universally enhance performance and may, in some cases, introduce contextual noise rather than beneficial guidance.

To complement the quantitative evidence provided by the confusion matrices, a brief qualitative examination of representative misclassifications was conducted. For instance, the headline "The company records operating losses despite increased revenues" was annotated as neutral in the ground truth but classified as negative by Gemini under few-shot prompting. This misclassification likely reflects the model's tendency to prioritize loss-related lexical cues over balanced financial indicators when demonstrations emphasize adverse outcomes. In contrast, the headline "Suspension of stock trading until further notice" was consistently predicted as negative by all models, despite being labeled as neutral in the dataset due to its regulatory and procedural nature within the Saudi market context. This observation illustrates how models may overinterpret regulatory actions as sentiment-bearing events, underscoring the inherent difficulty of distinguishing procedural disclosures from evaluative sentiment in financial news reporting.

Experiment 3 focused on the impact of stylistic variation in the prompt, using four different personas (P1–P4) to simulate diverse user roles (Fig. 10). The personas where tested against the LLMs and not PLMs due to their absence of instruction-tuned architectures. In this setting, ChatGPT demonstrated the highest adaptability, achieving an average accuracy of 91.14% across all personas, with peak performance under Persona 3 with 92.84% accuracy, 0.9282 F1-score, and 0.8732 Kappa. These results underscore ChatGPT's ability to generalize well across expressive prompt styles and reflect its strength in user-facing applications where language input is highly variable. Gemini maintained relatively stable performance across personas, with accuracies ranging from 85.92% to 87.18%, and minimal variation in agreement scores. While it did not achieve the highest metrics, its consistency suggests a degree of resilience to stylistic change, although it remained weaker than ChatGPT in terms of contextual nuance. Nuha, however, exhibited more volatility. Although it performed strongly under P1, its performance decreased in other conditions, particularly when neutral sentiment was present, suggesting a sensitivity to prompt tone and reduced stylistic generalization.

In response to RQ3, persona-driven prompting was found to exert a meaningful influence on sentiment classification performance, particularly in the case of ChatGPT. Assigning explicit financial roles—such as analyst or trader—provided contextual framing that aligned model reasoning with domain-specific decision perspectives, resulting in higher accuracy and stronger agreement with ground-truth labels. ChatGPT demonstrated the highest robustness across persona variations, suggesting that its instruction-tuning and reinforcement learning mechanisms support effective adaptation to role-based contextual cues. Gemini showed moderate improvement but remained sensitive to persona formulation, while Nuha exhibited greater variability, particularly in neutral sentiment recognition. Collectively, these results indicate that persona-based prompting can enhance the practical relevance of financial sentiment systems; however, its effectiveness depends on the model's ability to internalize and operationalize role-specific context.

The following analysis compares the effects of few-shot and persona-driven prompting on sentiment classification performance, as reported in Table III and IV. As shown in Table III, increasing the number of in-context examples does not lead to uniform performance gains across models. Nuha benefits from few-shot prompting, ChatGPT exhibits only marginal improvement, while Gemini experiences a notable decline. By contrast, Table IV indicates that persona-driven prompting yields more consistent and stable performance gains for instruction-tuned models, particularly ChatGPT, by aligning model reasoning with domain-relevant financial roles. Whereas few-shot prompting primarily affects how models infer task structure from demonstrations, persona-based framing appears to shape interpretation through contextual role expectations. Collectively, these findings suggest that contextual alignment achieved through role-based prompting may be more effective than increasing demonstration quantity alone for Arabic financial sentiment analysis, especially for models optimized for instruction-following behavior.

VI. Conclusion

This study presented a systematic empirical evaluation of Arabic pretrained language models and modern instruction-tuned large language models for financial sentiment analysis in Arabic news headlines within a Saudi market context. By examining model behavior across three experimental conditions—zero-shot classification, demonstration-based prompting, and persona-driven prompting—the study investigated how architectural design, training paradigms, and prompt formulation shape sentiment classification performance in a specialized financial domain.

The findings indicate that instruction-tuned large language models demonstrate greater flexibility and contextual sensitivity than Arabic pretrained language models when applied without task-specific fine-tuning. At the same time, the results reveal that the effectiveness of prompting strategies is strongly model-dependent. While Arabic-aligned models benefited from the inclusion of contextual demonstrations, certain general-purpose models achieved stronger performance under simpler zero-shot configurations and exhibited sensitivity to increased prompt complexity. These outcomes suggest that model scale alone does not ensure robustness, and that performance is critically influenced by the interaction between prompting strategy and training objectives.

Persona-driven prompting emerged as a particularly stable and effective mechanism for contextual alignment in financial sentiment analysis. Framing the task through role-specific perspectives—such as financial analysts or traders—enabled models to better interpret sentiment embedded within factual and regulatory financial language. This approach offers practical relevance for real-world financial applications, where sentiment interpretation is inherently shaped by decision context and stakeholder perspective rather than by generic polarity labels.

In conclusion, this study underscores the importance of aligning model selection and prompt design with domain-specific characteristics when deploying language models for financial text analysis. Rather than promoting a single universally superior model, the findings emphasize that effective Arabic financial sentiment analysis depends on the careful integration of linguistic context, model training paradigms, and prompt engineering. These insights contribute to ongoing scholarly discussions concerning the responsible and effective application of large language models in multilingual and domain-sensitive financial environments.

VII. Limitation and Future Works

This study has several limitations that should be acknowledged, which also highlight directions for future research. First, the dataset is constructed exclusively from Saudi financial news headlines obtained from Argaam, and therefore reflects the linguistic, regulatory, and institutional context of the Saudi market. While this specificity enhances contextual relevance, it restricts the generalizability of the findings to other Arabic financial environments, such as those in the UAE, Egypt, or the wider MENA region, where reporting practices and regulatory discourse may differ. Second, the analysis is limited to short news headlines, which often condense complex financial events into brief statements. Longer textual sources—such as full news articles, corporate reports, and social media content—may exhibit different sentiment dynamics and discourse structures, potentially influencing model performance and behavior. Third, although this study evaluates predictive performance, robustness, and prompting sensitivity, it does not incorporate interpretability techniques or user-centered validation. Such components are essential for understanding how financial practitioners interpret, trust, and operationalize AI-generated sentiment outputs.

Accordingly, future research should extend this work by constructing cross-market Arabic financial datasets, examining richer and more heterogeneous textual sources, and integrating explainability methods alongside human-in-the-loop evaluations. These directions would enable a more comprehensive assessment of both technical effectiveness and practical usability within real-world financial decision-support systems.

References

[1] C. Y. Lin and J. A. Lobo Marques, "Stock market prediction using artificial intelligence: A systematic review of systematic reviews," *Social Sciences & Humanities Open*, vol. 9, 2024.

[2] T. Phuoc, P. T. K. Anh, P. H. Tam, and C. V. Nguyen, "Applying machine learning algorithms to predict the stock price trend in the stock market – the case of Vietnam," *Humanities and Social Sciences Communications*, vol. 11, no. 1, 2024.

[3] G. Lombardo, M. Pellegrino, G. Adosoglou, S. Cagnoni, P. M. Pardalos, and A. Poggi, "Machine learning for bankruptcy prediction in the American stock market: Dataset and benchmarks," *Future Internet*, vol. 14, no. 8, p. 244, 2022.

[4] A. Bahatkar, A. Thakre, A. Kharde, A. Singh, and A. Nagdive, "Sentiment analysis of financial news article using TextCNN algorithm and web scrapping," in *International Conference on Emerging Trends in Engineering and Technology, ICETET*. IEEE Computer Society, 2025.

[5] P. Wattanakul and T. Phienthrakul, "Thai stock price prediction from daily news contents," in *Lecture Notes in Networks and Systems*, ser. Lecture Notes in Networks and Systems. Springer, 2022, vol. 191, pp. 851–861, iCTCS 2020.

[6] D. Valle-Cruz, V. Fernandez-Cortez, A. López-Chau, and R. Sandoval-Almazán, "Does Twitter affect stock market decisions? financial sentiment analysis during pandemics: A comparative study of the H1N1 and the COVID-19 periods," *Cognitive Computation*, vol. 14, no. 1, pp. 372–387, 2022.

[7] M. F. Hasan, V. S. Bhusari, L. R. Goranta, A. Dharmaraj, S. Kumar, and J. A. Dhanraj, "Artificial intelligence in financial management: Automating risk assessment and investment strategies," in *3rd International Conference on Advances in Computing, Communication and Materials, ICACCM 2024*. Institute of Electrical and Electronics Engineers Inc., 2024.

[8] N. Seong and K. Nam, "Predicting stock movements based on financial news with segmentation," *Expert Systems with Applications*, vol. 164, p. 113988, 2021.

[9] K. Du, F. Xing, R. Mao, and E. Cambria, "Financial sentiment analysis: Techniques and applications," *ACM Computing Surveys*, vol. 56, no. 9, 2024.

[10] B. Zhang, H. Yang, and X.-Y. Liu, "Instruct-FinGPT: Financial sentiment analysis by instruction tuning of general-purpose large language models," 2023. [Online]. Available: http://arxiv.org/abs/2306.12659

[11] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," 2019. [Online]. Available: http://arxiv.org/abs/1908.10063

[12] A. Buscemi and D. Proverbio, "ChatGPT vs Gemini vs LLaMA on multilingual sentiment analysis," 2024. [Online]. Available: http://arxiv.org/abs/2402.01715

[13] G. Fatouros, J. Soldatos, K. Kouroumali, G. Makridis, and D. Kyriazis, "Transforming sentiment analysis in the financial domain with ChatGPT," *Machine Learning with Applications*, vol. 14, p. 100508, 2023.

[14] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," 2020. [Online]. Available: http://arxiv.org/abs/2003.00104

[15] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The interplay of variant, size, and task type in Arabic pre-trained language models," 2021. [Online]. Available: http://arxiv.org/abs/2103.06678

[16] S. Aftan and H. Shah, "Using the AraBERT model for customer satisfaction classification of telecom sectors in Saudi Arabia," *Brain Sciences*, vol. 13, no. 1, 2023.

[17] S. Al-Khalifa, F. Alhumaidhi, H. Alotaibi, and H. S. Al-Khalifa, "ChatGPT across Arabic Twitter: A study of topics, sentiments, and sarcasm," *Data*, vol. 8, no. 11, 2023.

[18] A. Al-Thubaity, S. Alkhereyf, H. Murayshid, N. Alshalawi, M. Bin Omirah, R. Alateeq *et al.*, "Evaluating ChatGPT and Bard AI on Arabic sentiment analysis," 2023.

[19] A. Louati, H. Louati, E. Kariri, F. Alaskar, and A. Alotaibi, "Sentiment analysis of Arabic course reviews of a Saudi university using support vector machine," *Applied Sciences (Switzerland)*, vol. 13, no. 23, 2023.

[20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal *et al.*, "Language models are few-shot learners," 2020.

[21] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" 2022.

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[23] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li *et al.*, "A survey on in-context learning," 2024.

[24] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[25] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018.

[26] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," 2020.

[27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin *et al.*, "Training language models to follow instructions with human feedback," 2022.

[28] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai *et al.*, "Multitask prompted training enables zero-shot task generalization," 2022.

[29] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester *et al.*, "Finetuned language models are zero-shot learners," 2022.

[30] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus *et al.*, "Scaling instruction-finetuned language models," 2022.

[31] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[32] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," 2021.

[33] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," 2022.

[34] Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, and H. Ji, "Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration," 2024.

[35] M. Shanahan, K. McDonell, and L. Reynolds, "Role play with large language models," *Nature*, vol. 623, no. 7987, pp. 493–498, 2023.

[36] L. Salewski, S. Alaniz, I. Rio-Torto, E. Schulz, and Z. Akata, "In-context impersonation reveals large language models' strengths and biases," 2023.

[37] T. Loughran and B. McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, 2011.

[38] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.

[39] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán *et al.*, "Unsupervised cross-lingual representation learning at scale," 2020.

[40] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4996–5001.

[41] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, "XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization," 2020.

[42] X. V. Lin, T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig *et al.*, "Few-shot learning with multilingual language models," 2022.

[43] Argaam, "Argaam," https://www.argaam.com/, 2025. [Online]. Available: https://www.argaam.com/

Adaptive Differential Privacy in Federated Learning for Privacy-Preserving ICU Clinical Decision Support at the Edge

Moaad Almania[1]*, Anazida Zainal[2], Nurfazrina Binti Mohd Zamry[3],
Muhammad Haris[4], Ahmad Alnawasrah[5], Fuad A. Ghaleb[6]
Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, 81310, Malaysia[1]
College of Computing and Information Technology, Shaqra University, Shaqra, Kingdom of Saudi Arabia[1]
Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, 81310, Malaysia[2]
Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia[3]
Department of Information Communication Technology, British University of Bahrain, Bahrain[4]
College of Information Technology, Philadelphia University, Jordan[5]
College of Computing and Digital Technology, Birmingham City University, Birmingham, B47XG, United Kingdom[6]

***Abstract*—Clinical decision assistance that protects patient privacy is a prominent research issue in federated learning, edge computing, and intensive care analytics. However, typical differential privacy algorithms allocate equal privacy budgets to all clinical parameters regardless of sensitivity and predictive power. This may lower privacy-utility trade-offs. This study develops an adaptive differential privacy mechanism for federated learning that dynamically adjusts per-feature Laplace noise and privacy budgets to improve privacy protection while preserving clinical prediction performance. Adaptive Differential Privacy in Federated Learning (ADP-FL) dynamically adjusts per-feature Laplace noise and privacy budgets based on feature-scored sensitivity considering clinical relevance, statistical impact, and re-identification risk. The method was evaluated on the eICU Collaborative Research Database containing 113,265 ICU records and compared against Uniform Differential Privacy and Time-Adaptive Differential Privacy across different privacy budgets (ε). Experiments conducted on the eICU Collaborative Research Database achieved test accuracies of up to 82.7%. Across the evaluated privacy budgets, Feature-Adaptive Differential Privacy consistently outperformed both Uniform and Time-Adaptive Differential Privacy, achieving an F1-score of 0.6672 at $\varepsilon = 0.5$ while maintaining performance close to the no-DP baseline. These findings demonstrate that sensitivity-aware feature-level privacy budgeting preserves key predictive dimensions while minimizing unnecessary perturbation of less sensitive features, providing a robust foundation for privacy-compliant edge-based clinical decision support systems. The novelty of this study lies in its feature-scored adaptive differential privacy mechanism, which dynamically allocates per-feature privacy budgets and Laplace noise in federated learning for privacy-preserving edge-based clinical decision support.**

Keywords*—Federated learning; differential privacy; clinical decision support; edge computing; ICU data

I. Introduction

Clinical Decision Support Systems (CDSS) are increasingly used to support timely prediction, risk assessment, and treatment-related decision-making in intensive care environments [14]. In ICU settings, large volumes of physiological, laboratory, and treatment data are continuously generated across different hospitals and edge-based clinical systems. Although these data can improve predictive modelling, direct centralization of patient records raises serious concerns related to privacy, governance, and regulatory compliance. Federated Learning (FL) offers a practical solution by enabling multiple clinical sites or edge devices to train a shared model without transferring raw patient data to a central server [15].

Although FL provides this advantage, it does not ensure comprehensive privacy protection independently. There remains the potential for information to be revealed through exchanged model parameters or gradient modifications. This data could subsequently be utilized through membership inference, graph reconstruction, and property inference assaults [16]. In applications within the intensive care unit, these hazards become further pronounced as patient records often encompass highly sensitive clinical attributes. These factors encompass test biomarkers, respiratory state, vital signs, and indicators associated to severity. Therefore, privacy-preserving mechanisms are required to protect shared learning processes while maintaining acceptable clinical prediction performance.

Differential Privacy (DP) is widely used to provide formal privacy protection in machine learning and federated learning systems. However, many existing FL-based healthcare models apply uniform noise using a fixed global privacy budget. This approach treats all features as equally sensitive, although ICU variables do not carry the same level of disclosure risk or clinical importance. Certain metrics, including PaO_2, and creatinine concentrations, bilirubin levels, glucose measurements, dialysis status, and blood urea nitrogen (BUN), might yield more clinically pertinent insights than broad demographic or anonymized administrative variables. Imposing identical noise across all variables may inadvertently disrupt beneficial low-risk predictors, while simultaneously failing to adequately safeguard high-risk clinical characteristics [17].

Recent inquiries have been conducted regarding the adap-

*Corresponding author.

tive and individualized privacy strategies associated with federated learning. Examples of these concepts include adaptive gradient clipping, client-specific privacy budgeting, time-adaptive noise scheduling, and privacy-preserving frameworks for healthcare analytics [18]. Conversely, a considerable portion of these methodologies focuses on imaging datasets, general benchmark datasets, client-specific adaptation, or gradient-level privacy management [20]. Comparatively fewer studies have considered structured ICU tabular data, where each feature may have a different clinical meaning, predictive value, and re-identification risk. This creates a methodological gap for feature-sensitive privacy allocation in federated clinical decision support [19].

Healthcare-based FL studies have also shown that privacy protection often reduces predictive utility, especially when strict privacy budgets are used [21]. This privacy-utility trade-off is particularly important in ICU decision support because false negatives may delay clinical intervention, while excessive noise can weaken model reliability. Existing reviews on differential privacy in medical deep learning further indicate that model performance can decline considerably under strong privacy constraints, especially in heterogeneous and clinically complex datasets [27]. Therefore, there is a need for a privacy mechanism that can apply stronger protection to sensitive ICU features while preserving the predictive contribution of less sensitive but clinically useful variables.

Existing privacy-preserving FL approaches mainly emphasize uniform DP, gradient-level perturbation, time-adaptive noise, client-level adaptation, or medical imaging applications. Limited empirical evidence is available on feature-level adaptive privacy budgeting for structured ICU data. In particular, previous studies have not sufficiently addressed how privacy budgets can be dynamically allocated according to clinical relevance, statistical influence, and re-identification risk of individual ICU variables.

To address this gap, this study proposes Adaptive Differential Privacy in Federated Learning (ADP-FL) for privacy-preserving ICU clinical decision support at the edge. The primary purpose of this research is to create an adaptive differential privacy mechanism. This approach dynamically assigns privacy budgets and Laplace noise per feature based on the sensitivity of the ICU variables. The secondary goal is to assess the possibility of enhancing the privacy-utility trade-off over Uniform Differential Privacy and Time-Adaptive Differential Privacy with this privacy allocation that is sensitive to features. The proposed mechanism assigns per-feature privacy budgets and Laplace noise scales according to feature-scored sensitivity. Highly sensitive clinical variables receive stronger privacy protection through lower privacy budgets and greater perturbation, while less sensitive but predictive variables receive reduced perturbation to preserve model utility. The method is implemented using local client-side noise injection and evaluated on the eICU Collaborative Research Database containing 113,265 ICU records. ADP-FL is compared with Uniform Differential Privacy and Time-Adaptive Differential Privacy across different privacy budgets and gradient clipping norms using accuracy, F1-score, recall, and AUROC. This study contributes a feature-sensitive, lightweight, and governance-compatible privacy mechanism for federated ICU decision support systems [18].

II. Related Work

Federated learning in which privacy is preserved has garnered much research interest, and differential privacy is a key formal privacy guarantee. The novel DP-SGD approach introduced by Abadi, et al. [13] formed the foundation for gradient-level privacy in deep learning. Authors in [12] introduced cross-silo and cross-device FedAvg, which they showed for differential privacy constraints. Mironov [2] proposed Rényi Differential Privacy (RDP) which is a more suitable method for composition accounting for Gaussian processes.

Adaptive and customized privacy methods have been introduced to deal with the problems caused by uniform noise injection. Andrew, et al. [1] suggested adaptive gradient clipping in which the clipping value is learned from statistics collected from the previous iteration of the gradient. Fu, et al. [3] proposed an approach named Adap DP-FL that utilizes RDP accounting and client specific adaptability, with time-decaying noise schedules. In federated health care systems, Wu, et al. [4] examined the privacy-utility trade-offs and noted the need for a calibrated noise in non-IID settings.

Liu, et al. [5] explored the use of federated learning in healthcare, particularly with medical imaging and electronic health record (EHR) data. They pointed out the conflict between regulatory requirements (HIPAA, GDPR) and model efficacy. Saifullah, et al. [6] found that differing levels of privacy risk exist for various clinical traits and so there is a need for a differential level of sensitivity at the feature level. Recent comprehensive reviews have established that differential privacy is the most important privacy method in federated learning healthcare systems, but allocating uniform noise always leads to a decrease in the performance of the model compared to adaptive noise allocation strategies [25].

Adaptive differential privacy has recently been used in therapeutic applications. In order to detect COVID-19 using chest X-ray images, an adaptive differential privacy federated learning architecture was proposed by [7]. This platform is dynamically flexible in limiting privacy based on real-time data sensitivity assessment and shows exceptional performance in a wider range of healthcare settings. Pati, et al. [8] carried out a detailed survey on privacy preservation in federated learning (FL) in healthcare with a focus on gradient-level differential privacy (DP) mechanisms, which are more suitable for rare disease detection than parameter-level mechanisms, but also note that the budgeting distributed to different feature spaces is a significant challenge to address for sensitivity-preserving privacy in FL. A comprehensive review of 74 studies on DP in medical deep learning was performed by [9], which found that enforcing a strict privacy budget ($\varepsilon \sim 1$) often leads to a substantial drop in accuracy, especially in smaller and more diverse datasets. This highlights the need to develop and employ feature-adaptive noise allocation strategies, which are proposed in this study.

Previous work on differential privacy for the local setting [10] and PATE [11] mostly focuses on frequency estimation and semi-supervised contexts, both of which are less applicable to the context of continuous clinical feature regression.

III. Methodology

A. Dataset and Federated Configuration

1) Dataset: De-identified patient data from the United States' critical care units was used in the trials in the eICU Collaborative Research Database [19]. The database was accessed through the official eICU/PhysioNet data access pathway, and only de-identified records were used in accordance with the database usage requirements. There were 113,265 patient records with 21 binaries (intubated, vent, dialysis, medications), categorical (Glasgow Coma Scale: eyes, motor, verbal), and continuous (temperature, respiratory rate, heart rate, mean blood pressure, sodium, hematocrit, creatinine, BUN, glucose) attributes. The target variable was mechanical ventilation (vent) with a moderate class imbalance (73.18% negative cases [82,886 records] and 26.82% positive cases [30,379 records]). The target variable was excluded from the predictor set before model training to avoid label leakage [20].

2) Data preprocessing: The data set was preprocessed to ensure that the clinical variables could be used for model training prior to federated partitioning. Records were dropped if there were no target labels. For continuous variables, missing values were imputed using the median of the corresponding feature, while missing categorical values were imputed using the mode. All of the features were encoded as categorical variables, and the continuous variables were normalized using the Min-Max scaling method. This was done to ensure that all of the features were within a comparable range prior to the injection of differential privacy noise. The same pipeline of preprocessing was applied to the training data and tested on the rest of the test data that was held back. This action was taken to avert the loss of any data.

3) Federated configuration: The complete dataset, including 113,265 entries, was partitioned into training (90,612 entries) and testing (22,653 entries) subsets by stratified sampling. This was executed to preserve class ratios. The training records were distributed evenly across five simulated edge nodes, where each edge device is approximately 18,122 records. This is a rough approximation of a cross-silo healthcare federation with each site being an independent ICU or hospital unit. The weights of aggregation were calculated based on the population size of the local samples,

$$w_i = \frac{n_i}{N}, \tag{1}$$

with each node contributing approximately $w_i \approx 0.20$ under the uniform partitioning setting. The held-out test set was not used during local training or validation and was reserved only for final global model evaluation. Early pausing was implemented if the enhancement in validation accuracy fell below 0.1% for five consecutive communication rounds. The validation set was created from the training data using 20% of training and 80% of testing data [21].

B. Proposed ADP-FL Mechanism

1) Motivation for adaptive privacy in ICU federated settings: The rationale for Adaptive Privacy in ICU Federated Settings is compelling. Numerous privacy concerns arise while utilizing ICU data in federated contexts, which influenced the design of ADP-FL [23]. The attributes of clinical data disclosure vary significantly regarding disclosure risk: laboratory test results (PaO_2, creatinine, bilirubin) directly indicate the severity of organ dysfunction linked to patient identity and prognosis, whereas the lack of patient identity in de-identified demographic characteristics correlates with a substantially reduced likelihood of re-identification under HIPAA Safe Harbor [24]. Applying uniform noise to all characteristics fails to account for this variability, resulting in excessive perturbation of low-risk predictive features and reduced model utility, or insufficient protection of high-risk attributes and increased privacy vulnerability.

Secondly, shared gradient updates in federated learning training may reveal sensitive information if noise levels are insufficient, hence rendering the system vulnerable to membership inference and gradient inversion attacks [22]. Third, the proposed framework targets resource-constrained edge environments; therefore, an efficient privacy mechanism that supports local noise injection without requiring a trusted central curator is required. To address these challenges, ADP-FL dynamically determines the sensitivity of each feature to ensure the privacy of the data is maintained and it is usable across the federation [17].

2) ADP-FL overview: By adding on top of standard federated learning a feature-sensitive differential privacy mechanism, ADP-FL achieves a sensitivity score $S_{i,j} \in [0, 1]$ for every feature and then computes a per-feature privacy budget $\varepsilon_{i,j}$. Smaller $\varepsilon_{i,j}$ values, indicating stronger privacy and greater noise, are assigned to sensitive features, whereas bigger $\varepsilon_{i,j}$ values, indicating weaker perturbation and lower noise, are allocated to less sensitive features. Local Laplace noise is introduced at each edge device, followed by gradient aggregation post feature-specific budget calibration. This noise model is local client-side, meaning it does not necessitate a trusted server and may be utilized with secure aggregation methods. In clinical governance contexts, the Laplace composition of privacy is uncomplicated when employing pure composition [26].

Fig. 1 illustrates the comprehensive process of ADP-FL at an edge node, wherein per-feature sensitivity scores are employed to introduce feature-specific Laplace noise before transmitting the aggregated gradients to the central server. The server employs a strictly passive implementation that solely receives noisy merged gradients, computes weighted FedAvg, and transmits the updated global model [27].

3) Feature sensitivity taxonomy: Feature sensitivity scores were determined by integrating three criteria: 1) clinical relevance features critical to organ dysfunction assessment or treatment decision-making, 2) statistical influence features with high predictive impact on model gradients, and 3) re-identification risk features whose exposure could lead to patient re-identification under HIPAA Safe Harbor constraints [28].

Based on this assessment, features were grouped into three tiers:

- High-sensitivity ($S_j \approx 0.8$–1.0): PaO_2, creatinine, bilirubin, glucose, dialysis status, BUN, albumin, pH, $PaCO_2$. These features encode organ dysfunction

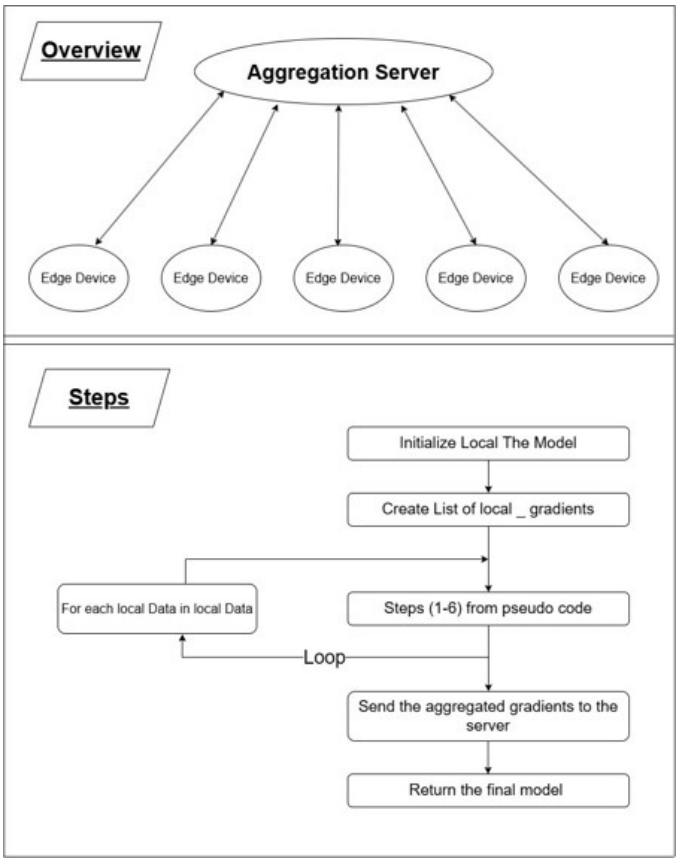


Fig. 1. ADP-FL: Adaptive DP with per-feature sensitivity, gradient clipping, Laplace noise at edge, FedAvg at server.

Algorithm 1: ADP-FL (Client-Side, Laplace Mechanism)

Input: ε_b, `sensitivity_scale`, clipping norm C, `local_data`, sensitivity scores S_j

For each communication round $t = 1, \ldots, T$**:**

1) Initialize `local_gradients` = [].
2) For each feature f_j in `local_data`:
   a) Compute per-feature budget:
   $$\varepsilon_j = \frac{\varepsilon_b}{S_j \times \texttt{sensitivity\_scale}}.$$
   b) Compute gradient:
   $$g_j = \nabla L(\theta_i).$$
   c) Clip:
   $$g_j^{\text{clip}} = \text{clip}(g_j, C).$$
   d) Add noise:
   $$g_j^{\text{dp}} = g_j^{\text{clip}} + \text{Laplace}\left(0, \frac{C}{\varepsilon_j}\right).$$
   e) Append g_j^{dp} to `local_gradients`.
3) Aggregate `local_gradients` and send to server.

Server: $\theta_{\text{global}} = \sum_i w_i \theta_i$, where $w_i = n_i/N$.

Privacy accounting: Total $\leftarrow \varepsilon_t + \varepsilon_t$ (pure-ε composition).

Fig. 2. Adaptive differential privacy in federated learning algorithm.

severity central to SOFA and APACHE scoring constructs and carry the highest clinical disclosure risk.

- Moderate-sensitivity ($S_j \approx 0.4$–0.7): Heart rate, mean blood pressure, respiratory rate, temperature, SpO_2, APACHE APS variables, GCS components (eyes, motor, verbal). These vital signs and derived ICU scores are important but less directly re-identifying.
- Low-sensitivity ($S_j \approx 0.1$–0.3): Age, gender, admission type, anonymized unit stay ID. De-identified demographic attributes under HIPAA Safe Harbor, with residual but lower re-identification risk.

4) Adaptive differential privacy in federated learning: The ADP-FL operation is executed locally at the edge devices during each communication round. The central server is limited to accepting aggregated noisy gradients, utilizing FedAvg, and transmitting the revised global model. No raw patient data, feature values, sensitivity scores, or utility proxies are transmitted to the central server. Fig. 2 illustrates the whole algorithm [29].

The ADP-FL approach executes operations on edge devices rather than on a central aggregation server. Edge devices construct local models, conduct local training, compute gradients, apply gradient clipping, and incorporate adaptive differential privacy noise, subsequently transmitting the noisy aggregated gradients to the server. To meet the strict local-edge threat model, this architecture ensures that patient data and sensitive metadata remain within the local edge device during processing [30].

The number of communication rounds (`NUM_COMM_ROUNDS`) is limited to 100 rounds, and the early stopping is based on the first time the validation accuracy improvement is below 0.1 per cent for five consecutive communication rounds. Aggregation weights w_i are proportional to local sample counts; under the equal distribution adopted here, each of the five nodes contributes $w_i \approx 0.20$.

The adaptive privacy operation can be summarized as follows. For each client i and feature j, a sensitivity score $S_{i,j}$ is first assigned. The privacy budget $\varepsilon_{i,j}$ is then adjusted according to the sensitivity level, where highly sensitive features receive smaller $\varepsilon_{i,j}$ values and therefore stronger Laplace perturbation. After local gradient clipping using clipping norm C, Laplace noise is applied before the noisy update is sent for FedAvg aggregation.

5) Parameter configuration: Table I summarizes the key parameters and their justification [16].

IV. Results and Discussion

A. Privacy-Accuracy Trade-off

Table II displays the test accuracy of the SGD Classifier across several privacy budgets $\varepsilon \in \{0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 1.5, 2.0\}$ and gradient clipping norms $C \in \{1.0, 3.0, 5.0, 10.0\}$, in comparison to a non-private Random Forest Classifier baseline (83.27%). The interval $[1, 10]$ is evaluated using the clipping norm C to mitigate the bias associated with excessive clipping and the introduced noise using recent privacy-budget and clipping-norm practices in differential privacy-based learning. Privacy budget assessment shows that the privacy-budget sweep $\varepsilon \in [0.1, 2.0]$ encompasses the privacy frameworks

TABLE I. ADP-FL PARAMETER VALUES AND JUSTIFICATION

Parameter	Value	Justification
ε_b	1.0	Baseline privacy budget balancing privacy and performance
Sensitivity scale	0.5–2.0	Scales noise to feature sensitivity, higher values apply more noise to sensitive features
Clipping norm C	1.0	Ensures gradient stability in federated updates, limits adversarial exploitation of large gradients
Learning rate η	0.01	Standard SGD step size for tabular clinical data
Mini-batch size	32	Balances computational efficiency and gradient stability
NUM COMM ROUNDS	100	With early stopping when accuracy improvement $< 0.1\%$ for 5 consecutive rounds

TABLE II. SGD CLASSIFIER TEST ACCURACY VS. PRIVACY BUDGET (ε) AND CLIPPING NORM (C)

$\varepsilon \backslash C$	$C = 1.0$	$C = 3.0$	$C = 5.0$
0.1	82.66%	81.81%	82.37%
0.2	82.51%	82.33%	81.76%
0.3	82.34%	81.45%	82.35%
0.5	81.88%	81.82%	82.60%
0.7	81.83%	81.75%	82.51%
1.0	82.47%	82.76%	81.93%
1.5	82.73%	82.55%	81.72%
2.0	81.75%	81.80%	82.62%

of both high-privacy and moderate-privacy regimes typically observed in practical applications [31].

Non-private Random Forest baseline: 83.27%. Noteworthy observations from Table II include the accuracy of the SGD Classifier fluctuating with different epsilon values, influenced by the selected clipping norm, as most clipping norms are sensitive to the noise-utility trade-off in differential privacy. Optimal accuracies are noted when epsilon falls between 0.5 and 1.5. This suggests that moderate privacy budgets establish an appropriate equilibrium between the incorporation of noise and the model's efficacy. Reducing the epsilon value, such as to 0.1, results in a rise in the added noise; nonetheless, the accuracy remains mostly unchanged, suggesting a robustness to differential privacy noise.

Standards for gradient clipping exert the following influence: The execution aligns with a Clipping Norm of 1.0, with a peak accuracy of 82.73 percent when ε is set to 1.5. The clipping norm of 3.0 strikes the optimal balance between privacy and precision, yielding the maximum accuracy of 82.76 percent when ε is configured to 1.0. The clipping thresholds of 5.0 and 10.0 yield a relatively diminished and more erratic accuracy, especially in the context of stringent privacy constraints. This occurs because increased clipping thresholds amplify the noise scaling in relation to C/ε and elevate the variance of the introduced perturbations. Employing diminished clipping norms ($C = 1.0, 3.0$) facilitates the optimization of the equilibrium between privacy safeguards and the reduction of the impact of introduced noise, resulting in diminished volatility across iterations and participants.

Comparison against the baseline: The baseline Random Forest Classifier, which does not use differential privacy, achieves an accuracy of 83.27%. When appropriately calibrated, the SGD Classifier with differential privacy can achieve accuracy comparable to its non-private equivalent (between 82.5% and 82.76%), demonstrating that privacy can be preserved without significant detriment to model efficacy, with a margin of less than one percentage point.

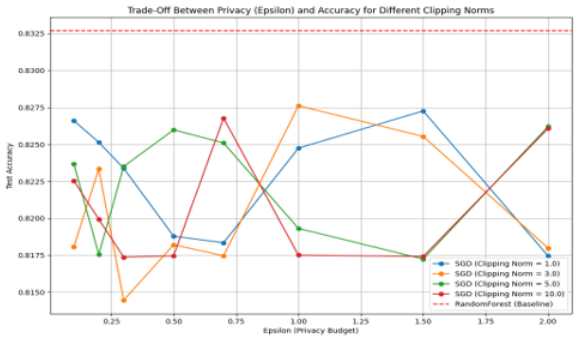


Fig. 3. Privacy-accuracy trade-off: SGDClassifier gradient clipping vs. RandomForest baseline.

Fig. 3 illustrates the trade-off between epsilon of the privacy budget and model accuracy over several gradient clipping norms. The optimal trade-off occurs at an epsilon range of 0.7 to 1.5 and a Clipping Norm of either 1.0 or 3.0, yielding an accuracy of roughly 82.7 while ensuring adequate privacy protection.

B. Dynamic Sensitivity Laplace vs. Baseline

This section compares a dynamic, feature-sensitivity-aware Laplace mechanism (ADP-FL) to a uniform differential privacy baseline spanning privacy budgets $\varepsilon \in \{0.2, 0.5, 1.0, 1.5, 2.0\}$. The model, training process, and evaluation split are constant, with the sole variation being the noise applied to training features after Min-Max scaling, while the test set remains unchanged. Table III presents the Accuracy, Precision, Recall, F1, and AUROC of both techniques at each ε.

Table III reports the controlled comparison between Dynamic (Feature-Adaptive) DP and Uniform DP using the same dataset, train–test split, federated-learning configuration, preprocessing, and evaluation protocol. Only the privacy-noise allocation strategy is changed between the two methods. Several significant tendencies can be discerned from Table III. Under the highest privacy level ($\varepsilon = 0.2$), the uniform strategy achieves superior metrics with an F1 score of 0.6540 and a Recall of 0.6363, compared to the dynamic strategy, which has an F1 score of 0.5114 and a Recall of 0.3754. This is consistent with the tenets of mechanism design: if ε is low, then the dynamic policy will also amplify noise on high-

TABLE III. PERFORMANCE ACROSS ε FOR DYNAMIC (ADAPTIVE) VS. UNIFORM DP

Strategy	i	ε	Accuracy	Precision	Recall	F1	AUROC
Dynamic	1	0.2	0.8076	0.8020	0.3754	0.5114	0.8256
Uniform	2	0.2	0.8194	0.6727	0.6363	0.6540	0.8276
Dynamic	3	0.5	0.8340	0.7343	0.5971	0.6586	0.8223
Uniform	4	0.5	0.8215	0.7063	0.5727	0.6326	0.8202
Dynamic	5	1.0	0.8342	0.7283	0.6093	0.6635	0.8276
Uniform	6	1.0	0.8194	0.7478	0.4933	0.5944	0.8219
Dynamic	7	1.5	0.8347	0.7256	0.6170	0.6669	0.8267
Uniform	8	1.5	0.8226	0.7265	0.5433	0.6217	0.8208
Dynamic	9	2.0	0.8351	0.7147	0.6414	0.6760	0.8269
Uniform	10	2.0	0.8223	0.7170	0.5576	0.6273	0.8280

sensitivity features, generating too much perturbation on the most clinically informative features and thus making it harder to detect minority groups. The AUROC scores are nearly the same (0.83) at $\varepsilon = 0.2$, meaning there has been very little loss of separability, with a decrease in performance primarily due to the balance of false negatives with respect to false positives. Table III reports the controlled comparison between Dynamic (Feature-Adaptive) DP and Uniform DP using the same dataset, train–test split, federated-learning configuration, preprocessing, and evaluation protocol. Only the privacy-noise allocation strategy is changed between the two methods.

As ε attains a moderate value, control over dominance will evidently shift to the dynamic mechanism. At $\varepsilon = 0.5$, ADP-FL exhibits an Accuracy of approximately 0.834 and an F1 score of approximately 0.6586, compared to approximately 0.8215 and approximately 0.6326 for uniform, with Recall similarly superior for dynamic (approximately 0.597 versus approximately 0.573). The improvement is evident at $\varepsilon = 1.0$, with dynamic metrics of Accuracy $= 0.834$ and F1 $= 0.664$, compared to the uniform baseline of 0.819 and 0.594. The Recall disparity is most pronounced at this operational point (0.609 versus 0.493). The trend persists until $\varepsilon = 1.5$ and $\varepsilon = 2.0$, during which the dynamic strategy maintains F1 within the 0.667–0.676 interval.

Fig. 4 illustrates a graph of Accuracy versus ε. The dynamic curve exhibits an ascent between 0.2 and 2.0, surpassing the uniform curve by around one to one-and-a-half percentage points at values of 0.5 and higher. The uniform curve is comparatively level near 0.82, indicating that the classifier utilizes indiscriminate noise to leverage more permissive privacy expenditure.

Fig. 5 illustrates the relationship between F1 and ε for dynamic and uniform noise. The F1 score under the dynamic method exhibits a monotonic improvement, increasing from around 0.511 at $\varepsilon = 0.2$ to approximately 0.676 at $\varepsilon = 2.0$. The uniform strategy is non-monotonic and has a notable minimum at $\varepsilon = 1.0$ to 0.594, just when the dynamic policy attains 0.664. This effect can be elucidated by analyzing Precision and Recall: dynamic Recall with $\varepsilon = 1.0$ is markedly superior (about 0.609 compared to approximately 0.493), but Precision is only somewhat worse (approximately 0.728 compared to approximately 0.748). The dynamic technique effectively reduces false negatives at ε values within the 0.5–2.0 range, albeit with a minor compromise in false positives concerning decision support.

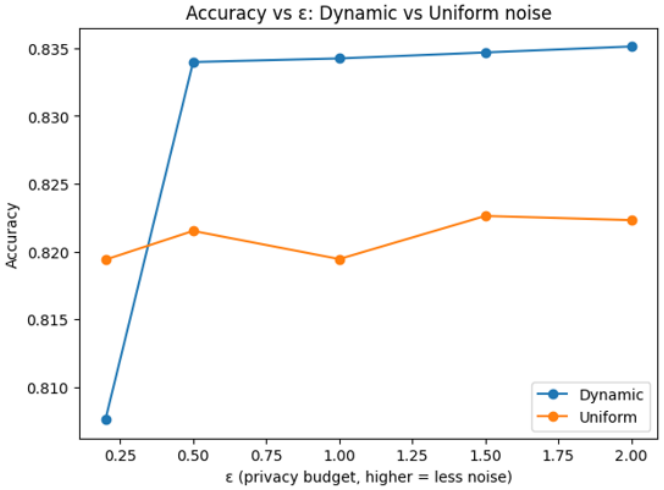


Fig. 4. Accuracy vs. ε for ADP-FL vs. Uniform DP.

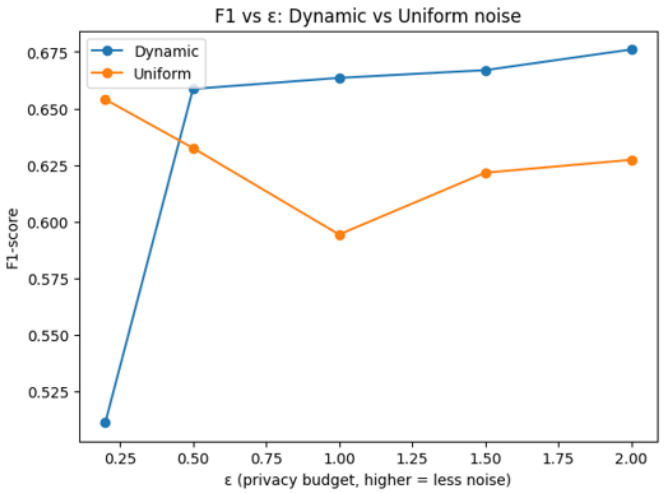


Fig. 5. F1-score vs. ε for dynamic (ADP-FL) vs. uniform DP.

Fig. 6 presents a detailed depiction of the aggregations of principal metrics at the value of $\varepsilon = 1.0$. The dynamic strategy exhibits superior Accuracy (0.834 vs. 0.819), Recall (0.609 vs. 0.493), and F1 score (0.664 vs. 0.594) at this mid-range operating point compared to the uniform baseline, despite a very small decline in Precision, which correlates with a shift

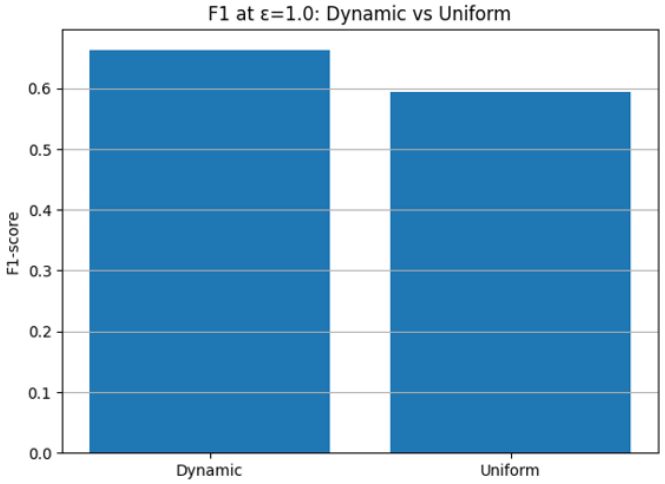


Fig. 6. Performance at $\varepsilon = 1.0$: Dynamic (ADP-FL) vs. uniform DP bar chart of accuracy, precision, recall, F1, and AUROC.

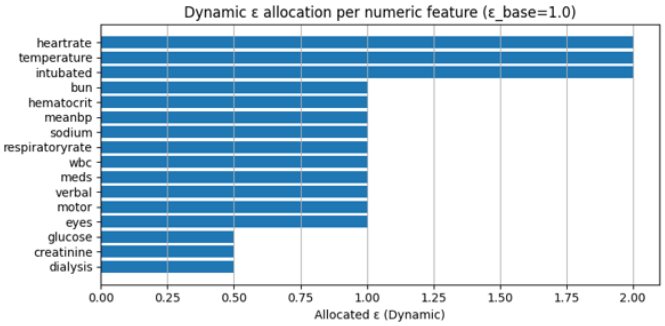


Fig. 7. Dynamic ε allocation per feature at $\varepsilon_b = 1.0$.

in focus towards sensitivity. This is the typical threshold used in safety applications where there are relatively low costs for dealing with false alarms and high costs for missing a positive indication.

Fig. 7 shows the per-feature ε of the dynamic policy for the case $\varepsilon_{\text{base}} = 1.0$, which explains the design rationale of the above results: clinical variables with high sensitivity (like PaO_2, creatinine, glucose) have a lower ε (higher noise); low sensitivity (like age, gender) have a higher ε (lower noise); middle sensitivity clinical variables (like vital signs) get an intermediate ε. This allocation rationale is corroborated by the results in Recall and F1 at moderate ε: the sensitivity-aware noise strategy is better at retaining valuable signals in less sensitive but still valuable features in comparison to a uniform policy.

C. Three-Way Privacy-Utility Comparison

Table IV presents an additional three-way comparison including Time-Adaptive DP; therefore, Table III is used as the primary controlled comparison between Feature-Adaptive DP and Uniform DP. Fig. 8 provide a comprehensive comparison of three differential privacy strategies: Uniform DP, Time-Adaptive DP, and Feature-Adaptive DP (ADP-FL), alongside a No-DP baseline (F1 $= 0.6668$) for privacy budgets $\varepsilon \in$

TABLE IV. F1-SCORE COMPARISON OF UNIFORM, TIME-ADAPTIVE, AND FEATURE-ADAPTIVE DP ACROSS ε. NO-DP BASELINE F1 $= 0.6668$

ε	Uniform DP	Time-Adaptive DP	Feature-Adaptive DP
0.1	0.4009	0.1458	0.4511
0.3	0.3584	0.3062	0.6011
0.5	0.2171	0.4642	0.6672
1.0	0.0724	0.4479	0.5118
2.0	0.4159	0.4919	0.5952

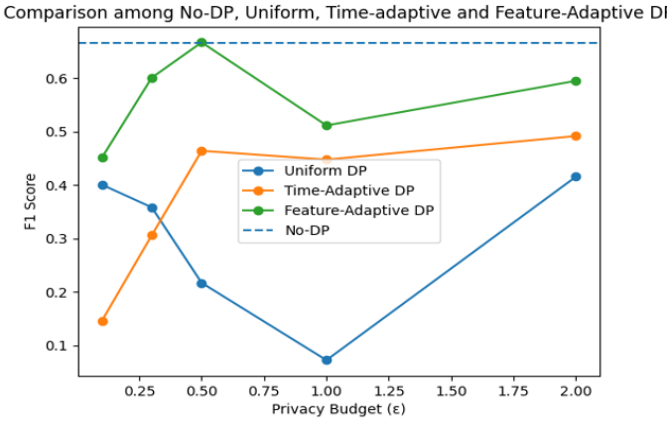


Fig. 8. F1-score comparison of No-DP, uniform DP, time-adaptive DP, and feature-adaptive DP.

$\{0.1, 0.3, 0.5, 1.0, 2.0\}$. Table IV illustrates a clear behavioral divergence among the three techniques. For the value of $\varepsilon = 0.1$, Feature-Adaptive DP achieves the highest F1 score of 0.4511 while Uniform DP comes in at 0.4009 and is much higher than Time-Adaptive DP with a score of 0.1458.

For $\varepsilon = 0.3$, the Feature-Adaptive DP performs significantly better (F1 $= 0.6011$ compared to 0.3584 in Uniform and 0.3062 in Time-Adaptive), showing that, under a moderate privacy budget, adaptive feature-level allocation is indeed much more effective at reducing degradation of minority classes resulting from noise disruption. When $\varepsilon = 0.5$, the score of Feature-Adaptive DP becomes 0.6672, which is nearly the same as the score of No-DP with the baseline score of 0.6668. In contrast, Uniform DP is drastically diminished (F1 $= 0.2171$), whereas TA DP is improved to 0.4642, albeit at a considerably lower level. The significant inadequacy of Uniform DP at $\varepsilon = 1.0$ (F1 $= 0.0724$) demonstrates that homogeneous noise calibration is unstable when it engages with non-uniform feature gradients. At $\varepsilon = 1.0$, Feature-Adaptive DP achieves F1 $= 0.5118$, and at $\varepsilon = 2.0$, all mechanisms converge to the No-DP baseline, Feature-Adaptive DP (0.5952) demonstrates superior performance.

Fig. 8 illustrates these trajectories. The FADP curve is maximized throughout all evaluated budgets, with the exception of the separate $\varepsilon = 0.2$, where uniform noise temporarily outperforms due to the compounding noise effect on high-sensitivity features. Practitioners in regulatory environments necessitating low ε should meticulously observe this crossover at the highest privacy thresholds.

The mechanistic rationale for the superiority of ADP-

FL is that high-sensitivity clinical features receive stronger perturbation, hence preserving the gradient signal that facilitates the memory of minority classes. Uniform differential privacy applies the same amount of noise to each feature, regardless of its predictive or privacy significance, leading to an overemphasis on informational gradients that adversely affect the imbalanced ventilation prediction task examined in this instance.

D. Literature-Level Comparison with Recent Published Studies

To meet the need of wider validation with the recent literature, Table V presents a comparison between the proposed ADP-FL model and the recently published federated learning studies for privacy, in the range of 2020–2026. The comparison reveals that recent work on healthcare FL has been successful; most of the studies concentrated on imaging data, privacy in blockchain, multimodal fusion, or general healthcare analytics, but did not include FL adaptive privacy budgeting for structured data in an ICU decision support system. The comparison indicates that the proposed ADP-FL model is not intended to simply maximize classification accuracy across different healthcare datasets; instead, its main contribution is to preserve ICU prediction performance while providing feature-sensitive privacy protection. Unlike blockchain-based and multimodal FL approaches, ADP-FL directly addresses the unequal privacy sensitivity of ICU variables by assigning different privacy budgets to different clinical features. This makes the proposed model more suitable for edge-based ICU clinical decision support where laboratory biomarkers, vital signs, and demographic attributes do not carry equal disclosure risk. Therefore, the new information offered by this study is not only improved internal performance over DP baselines, but also evidence that feature-level privacy allocation can preserve ICU prediction utility more effectively than uniform privacy treatment when clinical variables have unequal sensitivity and predictive importance. The performance values of the comparator studies in Table V are taken directly from their respective publications. Since these studies use different datasets, models, prediction tasks, and privacy settings, the comparison is literature-level rather than a direct head-to-head experimental benchmark.

The reported performance values should not be interpreted as a direct ranking of the methods because the studies use different datasets, prediction tasks, models, and privacy configurations. The contribution of ADP-FL is instead demonstrated through its controlled comparison with the privacy baselines under identical experimental conditions, where feature-sensitive privacy allocation preserves predictive utility while providing privacy protection for structured ICU data.

E. Operational Interpretation of Privacy-Utility Findings

For operational deployments, all evidence in favor of ADP-FL is within the range of $\varepsilon \in [0.5, 2.0]$. During this period, accuracy increases by approximately one percentage point or more when compared to uniform noise, while F1 also improves due to higher recall while maintaining a similar AUROC value. The operating margin is comparable with the typical ε of the order of few digits reported in current clinical machine learning pipelines. If the policy requires strong privacy ($\varepsilon = 0.1$–0.2), then the dynamic mechanism may over-perturb high-sensitivity features, which can reduce minority-class detection. In such cases, practitioners are advised to carefully consider the clinical miss-rate tolerance and privacy considerations.

The fact that privacy is pure-ε Laplace allows for simpler privacy accounting for clinical governance and regulatory reporting than (ε, δ)-DP techniques, and ADP-FL is well suited for installations regulated by HIPAA or GDPR, where audit trails require transparency in privacy accounting. The steady improvement of Recall with ADP-FL is important for use in the context of healthcare decision support in ICUs. The clinical cost of missing a mechanical ventilation warning is higher than the cost of a false alarm, and the dynamic control mechanism that keeps recall close to the No-DP baseline and at the same time ensures formal privacy is directly related to this cost imbalance.

F. Comparison with Alternative DP Approaches

When a trusted central aggregator or safe compute environment exists, Gaussian-noise DP-SGD may prove to be better suited to deeper models by altering the gradient-level noise based on the dynamic of the perturbation, instead of the scale of the features themselves. Standard smoothness assumptions give rise to an interesting pure-ε central differential privacy analog to objective perturbation in the convex learner case, e.g., logistic regression. Both strategies, however, presuppose a more reliable model than the local-edge danger model presented here, wherein data must be perturbed in situ prior to any uplink transmission.

Within the local-DP framework, the dynamic allocation of Laplace noise, which is responsive to variations in sensitivity, performed as the most suitable option in this study. Recent advancements in differential privacy for medical deep learning [27] suggest that other forms of DP, including generative modeling-based DP and hybrid federated designs, are forthcoming; nevertheless, these approaches have predominantly been evaluated using imaging data rather than structured tabular clinical data. The distinction between local Laplace and central Gaussian DP-SGD in terms of trust model and injection locus is not one of absolute superiority, but rather pertains to the trust model and the acceptability of central training: ADP-FL should be utilized when a trusted curator is not permitted, while Gaussian DP-SGD is likely to yield greater utility when central training is permissible.

For practical deployment feasibility assessment, the computational and communication features of ADP-FL were compared to FL without DP, Uniform DP-FL, and Time-Adaptive DP-FL. Additional overhead of ADP-FL is small as it does feature sensitivity scoring once before training and adds Laplace noise locally for each round of communication. Furthermore, the communication cost is similar to that of standard federated learning, where only updated noisy models are sent, and the original data, feature values, sensitivity scores, and feature-specific privacy budgets are not sent from the edge nodes.

G. Computational Overhead, Scalability, and Generalizability

To evaluate practical deployment feasibility, the computational and communication characteristics of ADP-FL were

TABLE V. LITERATURE-LEVEL COMPARISON WITH RECENT PRIVACY-PRESERVING FL APPROACHES

Reference	Application/Dataset	Privacy Approach	Reported Performance	Difference from Present Study
Proposed ADP-FL	eICU; 113,265 ICU records	Feature-adaptive DP with per-feature Laplace noise	Accuracy 82.76%; F1 = 0.6672 at $\varepsilon = 0.5$	Feature-level privacy allocation for structured ICU data.
Fu et al. [3]	MNIST and Fashion-MNIST	Adaptive clipping and time-decaying Gaussian noise with RDP	94.15% accuracy on MNIST and 79.56% on Fashion-MNIST at $\varepsilon = 2$	Adapts privacy across clients and rounds rather than individual clinical features.
Wu et al. [4]	CIFAR-10	Adaptive federated learning with differential privacy	81.25% accuracy (IID) and 79.05% (non-IID)	Uses adaptive DP for image classification rather than feature-sensitive ICU data.
Ahmed et al. [7]	COVID-19 chest X-ray images	Adaptive differential privacy federated learning	91.2% accuracy after 80 epochs with early stopping	Evaluated on medical images rather than structured ICU records.

compared with FL without DP, Uniform DP-FL, and Time-Adaptive DP-FL. Since ADP-FL performs feature-sensitivity scoring once before training and applies per-feature Laplace noise locally during each communication round, its additional overhead remains small. Moreover, the communication cost is comparable to standard federated learning because only noisy model updates are transmitted, while raw data, feature values, sensitivity scores, and feature-specific privacy budgets remain within the local edge nodes.

Table VI shows that for $\varepsilon = 0.5$, ADP-FL has the highest F1-score and other federated approaches have similar communication costs. The minor extra training cost is compensated by the better privacy-utility trade-off, as the proposed approach maintains performance close to the no-DP baseline. Additional multi-institutional and heterogeneous ICU datasets are needed, however, for further validation to prove the robustness and generalizability of ADP-FL in other real-world healthcare settings.

TABLE VI. PRACTICAL OVERHEAD AND UTILITY COMPARISON OF FEDERATED PRIVACY STRATEGIES

Method	Clients	Rounds	Relative Training Overhead	Relative Communication Cost	F1 at $\varepsilon = 0.5$
FL without DP	5	100	1.00×	1.00×	0.6668
Uniform DP-FL	5	100	1.01×	1.00×	0.2171
Time-Adaptive DP-FL	5	100	1.03×	1.00×	0.4642
Proposed ADP-FL	5	100	1.02×	1.00×	0.6672

V. CONCLUSION

This research presented and evaluated an adaptive differential privacy approach (ADP-FL) for federated learning in edge-based clinical decision assistance within the ICU. ADP-FL exhibits a markedly superior privacy-utility trade-off compared to Uniform DP and Time-Adaptive DP by dynamically allocating per-feature Laplace noise in accordance with clinically motivated sensitivity scores over the majority of privacy budgets. The novelty of this work is that it allocates feature-level privacy budgets based on clinical relevance, statistical influence, and re-identification risk, instead of having a uniform budget for all the ICU variables. ADP-FL demonstrates strong performance on the eICU database, achieving a test accuracy of 82.7%, closely aligning with the non-private baseline of 83.27%, and an F1-score of 0.6672 at $\varepsilon = 0.5$, nearly indistinguishable from the No-DP performance. These results demonstrate that the distribution of noise among the features is as significant as the quantity of noise applied. ADP-FL provides a governance-light, operational privacy-preserving edge Clinical Decision Support System (CDSS) implementation: pure-ε composition is audit-transparent, local noise injection requires no curator, and secure aggregation limits server involvement. The results showed that feature-sensitive privacy allocation is a better approach than uniform noise injection for ICU-based edge deployments, especially when clinical variables have varying disclosure risk and predictive value. The mechanism will grow in three ways in the future: 1) testing more complex classifiers while keeping the same privacy accounting; 2) creating or scheduling sensitivity weights instead of using fixed weights; and 3) combining privacy-budget accounting with edge resource limitations (latency, energy) to allow for a larger adaptive scheduling framework, which is at the heart of ADP-FL's privacy feature. Furthermore, to ensure the generalizability of the findings, further investigation is necessary to evaluate ADP-FL on more multi-institutional critical care unit datasets and across a broader spectrum of non-IID federated contexts. Future research may investigate if metaheuristic optimization methods like genetic algorithms and ant colony optimization can assist sensitivity weight scheduling.

ACKNOWLEDGMENT

The authors would like to thank to College of Computing and Information Technology, Shaqra University, Shaqra, Kingdom of Saudi Arabia for providing the funding to the student do research work in UTM.

REFERENCES

[1] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 17455–17466, 2021.

[2] I. Mironov, "Rényi differential privacy," in *Proc. 2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017, doi: 10.1109/CSF.2017.11.

[3] J. Fu, Z. Chen, and X. Han, "Adap DP-FL: Differentially private federated learning with adaptive noise," in *Proc. 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 656–663, 2022, doi: 10.1109/TrustCom56396.2022.00094.

[4] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022, doi: 10.1016/j.future.2021.09.015.

[5] H. Liu, C. Li, B. Liu, P. Wang, S. Ge, and W. Wang, "Differentially private learning with grouped gradient clipping," in *Proc. 3rd ACM International Conference on Multimedia in Asia (MMAsia)*, Art. no. 14, pp. 1–7, 2021, doi: 10.1145/3469877.3490594.

[6] S. Saifullah, D. Mercier, A. Lucieri, A. Dengel, and S. Ahmed, "The privacy-explainability trade-off: Unraveling the impacts of differential privacy and federated learning on attribution methods," *Frontiers in Artificial Intelligence*, vol. 7, Art. no. 1236947, 2024, doi: 10.3389/frai.2024.1236947.

[7] M. W. Rahman, M. R. Khan, and M. Nijim, "Privacy-preserving deep learning for disease diagnosis in medical imaging: A systematic review," *IEEE Access*, vol. 14, pp. 7248–7269, 2026, doi: 10.1109/ACCESS.2025.3647561.

[8] S. Pati, S. Kumar, A. Varma, B. Edwards, C. Lu, L. Qu, *et al.*, "Privacy preservation for federated learning in health care," *Patterns*, vol. 5, no. 7, Art. no. 100974, 2024, doi: 10.1016/j.patter.2024.100974.

[9] M. Mohammadi, M. Vejdanihemmat, M. Lotfinia, M. Rusu, D. Truhn, A. Maier, and S. Tayebi Arasteh, "Differential privacy for medical deep learning: Methods, tradeoffs, and deployment implications," *npj Digital Medicine*, vol. 9, Art. no. 93, 2026, doi: 10.1038/s41746-025-02280-z.

[10] N. T. Priya, V. S. Kumari, and K. J. Peter, "Evaluating the role of RAPPOR in strengthening data privacy within cybersecurity practices," in *Proc. 2024 International Conference on Communication, Computing, Smart Materials and Devices (ICCCSMD)*, pp. 1–5, 2024, doi: 10.1109/ICCCSMD63546.2024.11015118.

[11] Z. Qiao, P. Wang, P. Wang, Z. Ning, Y. Fu, Y. Du, *et al.*, "A dual-channel semi-supervised learning framework on graphs via knowledge transfer and meta-learning," *ACM Transactions on the Web*, vol. 18, no. 2, Art. no. 18, pp. 1–26, 2024, doi: 10.1145/3577033.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, *Proceedings of Machine Learning Research*, vol. 54, pp. 1273–1282, 2017.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016, doi: 10.1145/2976749.2978318.

[14] A. A. Bayor, J. Li, I. A. Yang, and M. Varnfield, "Designing clinical decision support systems (CDSS)—A user-centered lens of the design characteristics, challenges, and implications: Systematic review," *Journal of Medical Internet Research*, vol. 27, Art. no. e63733, 2025, doi: 10.2196/63733.

[15] S. Zhan, L. Huang, G. Luo, S. Zheng, Z. Gao, and H.-C. Chao, "A review on federated learning architectures for privacy-preserving AI: Lightweight and secure cloud–edge–end collaboration," *Electronics*, vol. 14, no. 13, Art. no. 2512, 2025, doi: 10.3390/electronics14132512.

[16] M. K. Quan, P. N. Pathirana, M. Wijayasundara, S. Setunge, D. C. Nguyen, C. G. Brinton, *et al.*, "Federated learning for cyber physical systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2025, doi: 10.1109/COMST.2025.3570288.

[17] X. Xu, Y. Li, L. Lu, and T. Wang, "ADPFL: Adaptive differential privacy-enhanced federated learning," *IEEE Internet of Things Journal*, vol. 12, no. 23, pp. 50682–50692, 2025, doi: 10.1109/JIOT.2025.3611410.

[18] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 17455–17466, 2021.

[19] J. Fu, Z. Chen, and X. Han, "Adap DP-FL: Differentially private federated learning with adaptive noise," in *Proc. IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2022, doi: 10.1109/TrustCom56396.2022.00094.

[20] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022, doi: 10.1016/j.future.2021.09.015.

[21] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific Reports*, vol. 12, no. 1, Art. no. 1953, 2022, doi: 10.1038/s41598-022-05539-7.

[22] J. Geiping, H. Bauermeister, H. Dröge, and M. Möller, "Inverting gradients—How easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems*, vol. 33, pp. 16937–16947, 2020.

[23] M. Almania, A. Zainal, F. A. Ghaleb, A. Alnawasrah, and M. Al Qerom, "Adaptive intrusion detection system with ensemble classifiers for handling imbalanced datasets and dynamic network traffic," *Journal of Robotics and Control*, 2025, doi: 10.18196/jrc.v6i1.23648.

[24] M. Almania, A. Zainal, F. A. Ghaleb, A. Alnawasrah, and M. Al Qerom, "Two-level feature selection for enhanced accuracy and reduced computational overhead in intrusion detection systems using rough set theory and binary particle swarm optimization," *Journal of Robotics and Control*, vol. 6, no. 1, pp. 262–271, 2025, doi: 10.18196/jrc.v6i1.23649.

[25] A. S. Khattak, A. B. Zain, R. B. Hassan, F. Nazar, M. Haris, and B. A. Ahmed, "Hand gesture recognition with deep residual network using sEMG signal," *Biomedical Engineering/Biomedizinische Technik*, vol. 69, no. 3, pp. 275–291, 2024, doi: 10.1515/bmt-2023-0208.

[26] S. Pati, S. Kumar, A. Varma, B. Edwards, C. Lu, Q. Le, J. J. Wang, A. Lakshminarayanan, S. H. Wang, M. J. Sheller, and K. Chang, "Privacy preservation for federated learning in health care," *Patterns*, vol. 5, no. 7, Art. no. 100974, 2024, doi: 10.1016/j.patter.2024.100974.

[27] M. Mohammadi, M. Vejdanihemmat, M. Lotfinia, M. Rusu, D. Truhn, A. Maier, and S. T. Arasteh, "Differential privacy for medical deep learning: Methods, tradeoffs, and deployment implications," *npj Digital Medicine*, vol. 9, Art. no. 93, 2026, doi: 10.1038/s41746-025-02280-z.

[28] K. H. Sayed, M. J. Cruyff, and P. G. M. van der Heijden, "Modeling evasive response bias in randomized response: Cheater detection versus self-protective no-saying," *Psychometrika*, vol. 89, no. 4, pp. 1261–1279, 2024, doi: 10.1007/s11336-024-10000-x.

[29] Z. Qiao, P. Wang, P. Wang, Z. Ning, Y. Fu, Y. Du, Y. Zhou, J. Huang, X.-S. Hua, and H. Xiong, "A dual-channel semi-supervised learning framework on graphs via knowledge transfer and meta-learning," *ACM Transactions on the Web*, vol. 18, no. 2, pp. 1–26, 2024, doi: 10.1145/3577033.

[30] W. Zhang, L. Wu, and S. Zhang, "Clinical phenotype of ARDS based on K-means cluster analysis: A study from the eICU database," *Heliyon*, vol. 10, no. 20, Art. no. e39198, 2024, doi: 10.1016/j.heliyon.2024.e39198.

[31] S. M. Orthi, M. H. Rahman, K. B. Siddiqa, M. Uddin, S. Hossain, A. Al Mamun, and M. N. Khan, "Federated learning with privacy-preserving big data analytics for distributed healthcare systems," *Journal of Computer Science and Technology Studies*, vol. 7, no. 8, pp. 269–281, 2025, doi: 10.32996/jcsts.2025.7.8.31.

An Efficient and Reliability-Oriented Deep Learning Framework for Alzheimer's MRI Classification

Yashwant A.*, Rajaprakash S.
Department of Computer Science and Engineering,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai 600062, Tamil Nadu, India

***Abstract*—Reliable Alzheimer's disease (AD) magnetic resonance imaging (MRI) classification requires evaluation beyond accuracy alone. This study investigates a four-class image-level framework based on ImageNet-pretrained ResNet-18 using 6,400 public MRI images labelled Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. Original images were assigned to seventy percent training, fifteen percent validation, and fifteen percent test partitions before augmentation. The selected no-CLAHE pipeline used training-only transformations, 128 by 128 pseudo-RGB inputs, channel normalization with mean and standard deviation of 0.5, a 512 to 256 to 4 classifier, class-weighted cross-entropy with label smoothing, AdamW optimization, and five fixed seeds. Mean accuracy was 98.38 percent with a standard deviation of 0.38 percent, balanced accuracy was 98.86 percent with a standard deviation of 0.40 percent, macro F1-score was 90.73 percent with a standard deviation of 1.79 percent, and macro-AUC was 0.9993 with a standard deviation of 0.0009. For seed 42, validation-fitted temperature scaling reduced negative log-likelihood, Brier score, and expected calibration error without changing predicted labels. The model contains 11.31 million trainable parameters and achieved 3.08 ms batch-1 inference on a Tesla T4. Grad-CAM supported qualitative inspection. The results establish a reproducible, confidence-aware, perturbation-evaluated, and resource-efficient image-level framework within the stated public-dataset protocol.**

***Keywords*—Alzheimer's disease; magnetic resonance imaging; ResNet-18; confidence calibration; robustness; Grad-CAM; computational efficiency**

I. Introduction

Alzheimer's disease (AD) is a major cause of dementia, and structural magnetic resonance imaging (MRI) is widely used to investigate disease-related brain changes. Deep learning has been increasingly applied to medical-image representation learning and neuroimaging classification [1]–[3]. In healthcare analytics, however, a reported performance value is meaningful only when the data unit, split protocol, class prevalence, preprocessing sequence, and separation between validation and test data are clearly described. These considerations are especially important for public image collections in which subject identifiers and acquisition metadata may be unavailable [3], [4].

Residual networks use skip connections to facilitate optimization of convolutional models [5]. An ImageNet-pretrained ResNet-18 can be adapted to pseudo-RGB MRI inputs through transfer learning [6], [7]. In imbalanced medical-image collections, training-only augmentation, class-aware loss functions, and macro-averaged evaluation may provide more useful information than accuracy alone [8], [9].

Contrast-limited adaptive histogram equalization (CLAHE) was considered as a candidate local-contrast transformation because adaptive histogram equalization improves local contrast and CLAHE limits excessive local histogram amplification [10], [11]. This rationale does not imply that CLAHE must improve an AD-MRI classifier. Contemporary MRI-based AD studies use heterogeneous image representations, architectures, and validation designs [12]–[17]; preprocessing choices should therefore be evaluated on the target collection rather than assuming that they are beneficial *a priori*.

This work evaluates a reliability-oriented and resource-efficient ResNet-18 framework for the public labels Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. In this study, "reliability-oriented" refers to the combined use of five-seed reproducibility, probability calibration, selective prediction, controlled perturbation testing, qualitative model inspection, and computational profiling as complementary technical checks within the evaluated protocol. The selected configuration preserves the resized grayscale intensity structure and uses online training augmentation. The study makes six contributions:

- It documents a fixed stratified image-level 70/15/15 partition of 6,400 original images before stochastic transformation and reports five-seed stability.
- It addresses substantial class imbalance through class-weighted optimization and class-sensitive held-out metrics, with explicit reporting of the nine-image Moderate Demented test subset.
- It reports a transparent configuration-level comparison between no CLAHE with online augmentation and CLAHE without augmentation, without claiming isolated causal effects.
- It applies Grad-CAM to a representative 0.5/0.5 normalization checkpoint for qualitative model inspection while avoiding anatomical or clinical localization claims.
- It evaluates validation-fitted temperature scaling and confidence-ranked selective prediction to characterize probability reliability and risk–coverage behaviour.
- It quantifies sensitivity to controlled image perturbations and directly measures parameters, multiply–accumulate operations, latency, throughput, checkpoint size, and peak inference memory.

*Corresponding author.

II. Related Work and Motivation

A. Deep Learning for AD MRI Classification

MRI-based AD classification has been investigated using 2D and 3D convolutional networks, transfer-learning methods, multi-slice systems, attention mechanisms, and transformer-based models [12]–[17]. Direct numerical comparison across these studies is difficult because cohorts, image units, imaging protocols, label definitions, class prevalences, and validation regimes vary. A model tested on independently partitioned participants or complete volumes is not directly comparable with a model tested on individual images from a public folder structure.

Wen *et al.* emphasized the importance of carefully defined validation protocols for reproducible MRI-classification studies [3]. Ebrahimighahnavieh *et al.* similarly highlighted the value of clearly specified preprocessing and evaluation procedures in neuroimaging deep learning [4]. Accordingly, this study reports reproducible image-level experimentation and retains the public dataset labels as supplied, while keeping the interpretation aligned with the evaluated protocol.

B. Online Augmentation and Class-Sensitive Evaluation

Data augmentation introduces training-time variation without altering the held-out test set. Geometric and photometric perturbations may reduce overfitting, but they should remain plausible for the target task and must not be applied before a split intended to represent unseen data [8]. Here, horizontal flips, rotations limited to $\pm 10^{\circ}$, and modest brightness–contrast jitter are applied only to training images.

Class imbalance affects both learning and evaluation. Moderate Demented comprises only 64 of 6,400 images. A high overall accuracy can therefore coexist with class-dependent variation in precision and F1-score. Class-weighted loss and macro-averaged reporting provide complementary information for this imbalanced setting [9]. ROC curves can remain visually favorable when prevalence is low; precision–recall curves and class-specific precision are consequently reported together with macro-AUC [18], [19].

C. Rationale for CLAHE Screening

CLAHE was evaluated as a candidate preprocessing operation, not as a presumed source of performance gain. It locally redistributes grayscale contrast while restricting the over-amplification associated with standard adaptive histogram equalization [10], [11]. In structural MRI, local intensity variation may influence anatomical representation, while enhancement may also modify intensity relationships learned by a convolutional model. CLAHE was therefore screened empirically during pipeline selection. The final selected configuration retained the resized grayscale representation without CLAHE.

The comparison documents the practical configuration-selection decision. Because CLAHE use and online augmentation changed jointly, the observed difference is reported as a pipeline-level comparison rather than attributed to either component in isolation.

D. Calibration, Selective Prediction, and Robustness

Classification accuracy and confidence calibration describe complementary properties of predictive performance. Temperature scaling provides a simple post-hoc calibration method in which a single validation-fitted scalar rescales logits without changing the model weights or predicted class ordering [26]. Selective prediction complements calibration by examining the trade-off between retained coverage and prediction risk when low-confidence cases are withheld [27]. Robustness to common image corruptions is also distinct from clean-set discrimination, and controlled perturbation tests can reveal sensitivity to noise, blur, brightness, contrast, and rotation [28]. In this study, these analyses are applied to the unchanged representative seed-42 checkpoint and are interpreted as complementary technical reliability checks within the evaluated dataset and protocol.

III. Materials and Methods

A. Dataset and Study Scope

The MRI image collection was obtained from Mendeley Data [20]. It contained 6,400 images in four labelled folders: Non-Demented (3,200), Very Mild Demented (2,240), Mild Demented (896), and Moderate Demented (64). The dataset is a public image-classification collection, and the present study therefore reports its results using the supplied image-level organization. Thus, the experimental unit in this study is an image, and all reported performance values are interpreted at the image level.

No new human participants were recruited, no intervention was performed, and no identifiable clinical records were accessed. The directory terminology is retained exactly from the public collection; in particular, Moderate Demented is treated as a dataset class label rather than as a newly proposed clinical staging system. Table I summarizes the class composition and fixed image-level partition, while Fig. 1 shows the corresponding class-support profile.

B. Split Integrity and Evaluation Workflow

Original image records were assigned to a fixed stratified 70/15/15 split: 4,480 training images, 960 validation images, and 960 test images. The same saved split manifest was reused for every condition and seed. No stochastic transformation was applied to validation or test images. This design prevents a newly generated transformation of an image from entering a different partition during the evaluated training process.

Fig. 2 summarizes the workflow. The held-out test set remains fixed while training is repeated with seeds 42, 52, 62, 72, and 82. Variation across seeds therefore represents training stochasticity under one shared image-level partition and is reported specifically as a measure of training reproducibility.

C. Image Representation and Training Transformations

Each image was converted to grayscale, resized to 128×128 pixels using bilinear interpolation, and replicated across three channels to form a pseudo-RGB input. The selected configuration retained the resized grayscale intensities and applied channel-wise normalization with mean and standard deviation equal to 0.5:

TABLE I. DATASET COMPOSITION AND FIXED IMAGE-LEVEL PARTITION. THE SPLIT WAS CREATED BEFORE ONLINE AUGMENTATION

Class	Total	Training	Validation	Test	Test support (%)
Non-Demented	3,200	2,240	480	480	50.00
Very Mild Demented	2,240	1,568	336	336	35.00
Mild Demented	896	627	134	135	14.06
Moderate Demented	64	45	10	9	0.94
Total	6,400	4,480	960	960	100.00

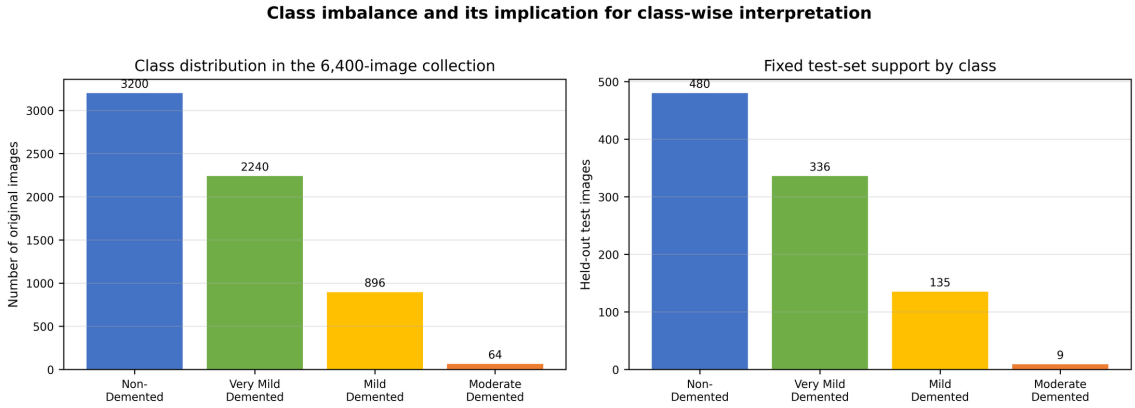


Fig. 1. Class prevalence in the public collection and fixed held-out test support. The Moderate Demented support is shown explicitly to contextualize precision, F1-score, and class-level performance differences.

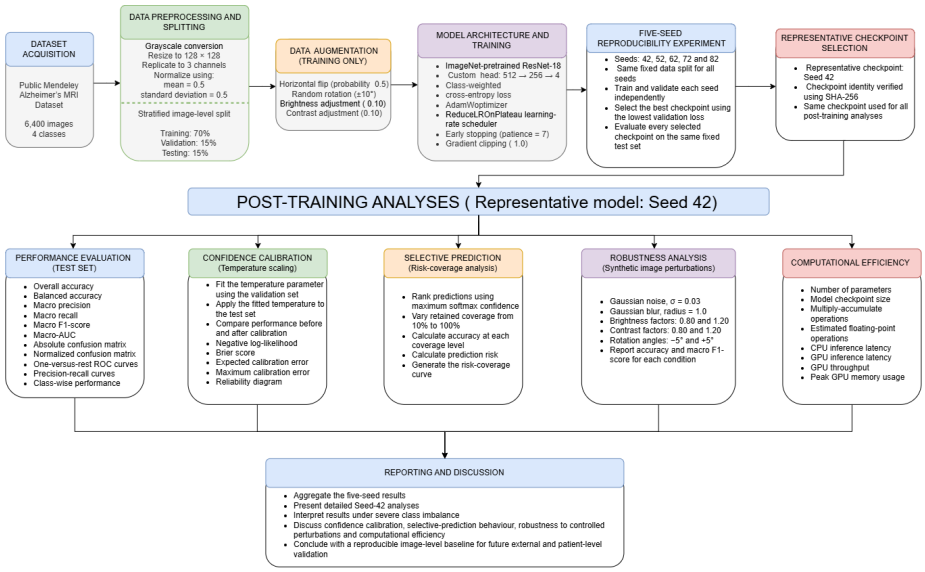


Fig. 2. Reproducible image-level evaluation workflow. Original files are split before training transformations; validation loss selects the checkpoint, and the unchanged held-out test partition is evaluated across five fixed seeds.

$$\widetilde{x}_c = \frac{x_c - 0.5}{0.5}, \qquad c \in \{1, 2, 3\}. \tag{1}$$

Only training images received random horizontal flipping with probability 0.5, random rotation in $[-10°, 10°]$, and brightness–contrast perturbation with magnitude 0.10. These transformations were re-sampled online and never stored as offline copies. Fig. 3 shows the partition composition, while Fig. 4 illustrates the candidate CLAHE operation.

D. Model Architecture, Optimization, and Selection

The backbone was ImageNet-pretrained ResNet-18 [5]. ResNet-18 was selected because its compact residual architecture provides a practical balance between representational capacity, transfer-learning support, and computational cost for a lightweight image-level framework. Its final layer was replaced by a task-specific head comprising a 512-to-256 fully connected layer, batch normalization [21], ReLU, dropout with rate 0.30 [22], and a 256-to-4 output layer. The model contains 11,309,380 trainable parameters. Table II summarizes



Fig. 3. Fixed train, validation, and test composition by class. The partition was created before online augmentation.

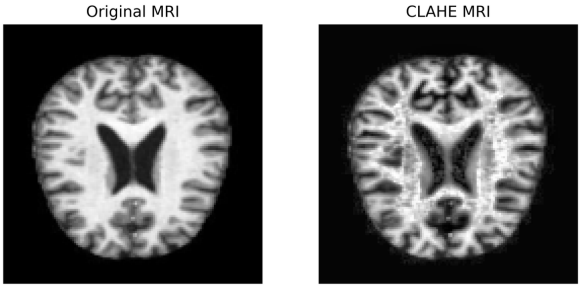


Fig. 4. Illustrative original and CLAHE-transformed grayscale MRI image. CLAHE was screened as a literature-motivated local-contrast candidate and was not retained in the final selected configuration.

the selected ResNet-18 training configuration.

TABLE II. SELECTED RESNET-18 TRAINING CONFIGURATION

Item	Setting
Input	$3 \times 128 \times 128$ pseudo-RGB image
Backbone	ImageNet-pretrained ResNet-18
Classifier head	$512 \to 256 \to 4$; batch normalization, ReLU, dropout 0.30
Trainable parameters	11,309,380
Loss	Class-weighted cross-entropy; label smoothing 0.05
Optimizer	AdamW [23]
Initial learning rate	10^{-4}
Weight decay	10^{-4}
Scheduler	ReduceLROnPlateau; factor 0.5, patience 2
Batch size	16
Epoch limit	25
Model selection	Lowest validation-loss checkpoint
Early stopping	Patience 7
Seeds	42, 52, 62, 72, 82

The training loss used class weights inversely proportional to observed training support:

$$w_k = \frac{N}{K n_k}, \tag{2}$$

where, N is the number of training images, $K = 4$ is the number of classes, and n_k is the training support of class k. Weighted cross-entropy was combined with label smoothing of 0.05 [24]. AdamW used initial learning rate 10^{-4} and weight decay 10^{-4} [23]. A ReduceLROnPlateau scheduler halved the learning rate after two non-improving validation-loss epochs. Gradient clipping at 1.0 and early stopping with patience 7 were used. The checkpoint with the lowest validation loss was restored before test inference.

E. Evaluation Measures and Statistical Reporting

For each seed, accuracy, balanced accuracy, macro precision, macro recall, macro F1-score, macro one-versus-rest AUC, and per-class precision, sensitivity, specificity, and F1-score were computed on the fixed test partition. In the multi-class setting, balanced accuracy is the average class recall:

$$\text{Balanced Accuracy} = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FN_k}. \tag{3}$$

The macro F1-score is the unweighted average of class-specific F1-scores:

$$F1_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K}\frac{2P_k R_k}{P_k + R_k}. \tag{4}$$

Metrics are reported as mean $\pm$ standard deviation across five seeds. This summarizes variability due to initialization, data-loader order, and stochastic augmentation under the same split. Accordingly, the five-seed summary is interpreted as training reproducibility under the fixed image-level protocol rather than as a population-level estimate.

F. Configuration-Level Screening

CLAHE was evaluated as a candidate local-contrast operation, whereas online augmentation introduced limited training-time variation. The final no-CLAHE configuration with online augmentation was compared with a CLAHE-based configuration without online augmentation. Both conditions shared the fixed split, ResNet-18 architecture, 0.5/0.5 normalization, class-weighted loss, optimizer, scheduler, batch size, seed set, and checkpoint policy. Since contrast enhancement and augmentation changed jointly, this screening comparison identifies the stronger evaluated pipeline and is interpreted as a configuration-level comparison without assigning the observed difference to either component in isolation.

G. Explainability Analysis Using Grad-CAM

Gradient-weighted class activation mapping (Grad-CAM) [25] was applied to the final convolutional layer of the last ResNet-18 residual block (`layer4[-1].conv2`) to provide qualitative insight into the spatial evidence associated with model predictions. The analysis used the representative seed-42 checkpoint trained with no CLAHE, training-only online augmentation, and the 0.5/0.5 channel normalization defined in Eq. (1). For each class, one correctly classified held-out test image with confidence closest to the class-specific median among correct predictions was selected to reduce reliance on only the most confident examples. Gradients were calculated

with respect to the predicted class, and each activation map was rectified, normalized to the interval $[0, 1]$, resized to the input resolution, and overlaid on the corresponding grayscale MRI image. These maps are qualitative explanations of model attention and are not interpreted as anatomical, pathological, or clinical localization. The visualizations preserve the orientation supplied by the public image collection; no additional anatomical orientation markers or slice-index metadata are introduced.

H. Confidence Calibration and Selective Prediction

The reliability analysis used the representative seed-42 checkpoint with the 0.5/0.5 channel normalization defined in Eq. (1). A single temperature parameter $T > 0$ was fitted by minimizing cross-entropy on the fixed validation partition only [26]. Test probabilities were then calculated as:

$$p(y = k \mid x) = \frac{\exp(z_k/T)}{\sum_{j=1}^{K} \exp(z_j/T)}, \tag{5}$$

where, z_k denotes the logit for class k. Calibration was evaluated on the unchanged test partition using negative log-likelihood (NLL), multiclass Brier score, expected calibration error (ECE), and maximum calibration error (MCE). ECE and MCE used 15 equal-width confidence bins. Because scalar temperature scaling is monotonic in the logits, it changes probability sharpness but not the predicted class labels.

Selective prediction was evaluated by ranking test predictions according to maximum softmax confidence and retaining progressively larger coverage fractions from 10% to 100% [27]. Selective risk was defined as one minus accuracy among retained predictions. Macro F1-score was recorded but was not emphasized at very low coverage because some classes may be absent from the retained subset.

I. Robustness Stress-Test Protocol

The unchanged seed-42 checkpoint was evaluated under deterministic synthetic perturbations applied only at test time. The conditions were Gaussian noise with standard deviation 0.03, Gaussian blur with radius 1.0, brightness factors of 0.80 and 1.20, contrast factors of 0.80 and 1.20, and rotations of -5° and $+5^\circ$. After perturbation, images followed the same 128×128 pseudo-RGB conversion and 0.5/0.5 normalization used for the clean test set. Accuracy, balanced accuracy, macro precision, macro recall, macro F1-score, and macro-AUC were computed for every condition. These tests quantify sensitivity to the specified controlled input changes [28]; broader scanner-, site-, and acquisition-related variation is reserved for future multi-centre evaluation.

J. Computational Efficiency Evaluation

Computational demand was measured using trainable parameter count, checkpoint size, multiply–accumulate operations (MACs), estimated floating-point operations (FLOPs), inference latency, throughput, and peak allocated GPU memory. MACs were profiled for one $3 \times 128 \times 128$ input, with FLOPs approximated as twice the MAC count. Latency measurements used repeated warm-up and timed forward passes. CPU measurements used batch sizes 1 and 16; Tesla T4 GPU measurements used batch sizes 1 and 32. These values are reported as an implementation profile for the stated hardware, batch sizes, and runtime conditions.

IV. Results

A. Aggregate Configuration Comparison

The selected no-CLAHE/augmentation configuration achieved 98.38% ± 0.38% accuracy, 98.86% ± 0.40% balanced accuracy, 86.94% ± 1.79% macro precision, 90.73% ± 1.79% macro F1-score, and 0.9993 ± 0.0009 macro-AUC. The CLAHE/no-augmentation configuration achieved 97.19% ± 0.32% accuracy, 97.95% ± 0.23% balanced accuracy, 82.56% ± 0.66% macro precision, 86.04% ± 0.76% macro F1-score, and 0.9980 ± 0.0016 macro-AUC. The selected configuration showed a 1.19 percentage points higher accuracy and a 4.68 percentage points higher macro F1-score than the alternate configuration under the same five-seed image-level protocol (Fig. 5).

B. Seed-Level Stability

Table IV lists held-out metrics for all five selected-configuration runs. Accuracy ranged from 98.02% to 98.96%, while macro F1-score ranged from 87.98% to 92.66%. Fig. 6 shows seed-level accuracy and macro F1-score. Seed variation describes training stochasticity under one fixed image-level split and therefore provides a focused measure of training reproducibility.

C. Class-Wise Performance

Table V reports class-wise results for the selected configuration. Non-Demented, Very Mild Demented, and Mild Demented achieved high mean F1-scores. Moderate Demented achieved 100% mean sensitivity across seeds, with 49.89% ± 7.64% mean precision and 66.29% ± 6.88% mean F1-score. This class-wise pattern shows why the 98.38% overall accuracy should be interpreted together with macro-level and per-class metrics. The nine-image Moderate Demented test support is therefore reported explicitly so that the sensitivity, precision, and F1-score can be interpreted in the context of the available class support.

Fig. 7 complements Table V. The left panel compares per-class F1-scores under the two configurations. The right panel shows the selected configuration's precision–sensitivity relationship. The Moderate Demented class combines high sensitivity with lower precision, making macro-level and class-wise reporting especially informative alongside overall accuracy.

D. Representative Learning and Discrimination Analyses

Fig. 8 presents training and validation loss and macro F1-score for the representative seed-42 selected run. The lowest validation-loss checkpoint was retained for test inference. Fig. 9 presents one-versus-rest ROC and precision–recall curves for the same run. ROC and PR views are both included because class support is unequal; numerical claims remain the five-seed aggregate results in Table III.

TABLE III. PIPELINE-SCREENING COMPARISON ON THE FIXED HELD-OUT TEST SET. VALUES ARE MEAN ± STANDARD DEVIATION OVER FIVE FIXED TRAINING SEEDS. THE PIPELINES DIFFER JOINTLY IN CLAHE USE AND ONLINE AUGMENTATION

Configuration	Accuracy (%)	Balanced accuracy (%)	Macro precision (%)	Macro recall (%)	Macro F1-score (%)	Macro-AUC
No CLAHE + augmentation	98.38 ± 0.38	98.86 ± 0.40	86.94 ± 1.79	98.86 ± 0.40	90.73 ± 1.79	0.9993 ± 0.0009
CLAHE + no augmentation	97.19 ± 0.32	97.95 ± 0.23	82.56 ± 0.66	97.95 ± 0.23	86.04 ± 0.76	0.9980 ± 0.0016

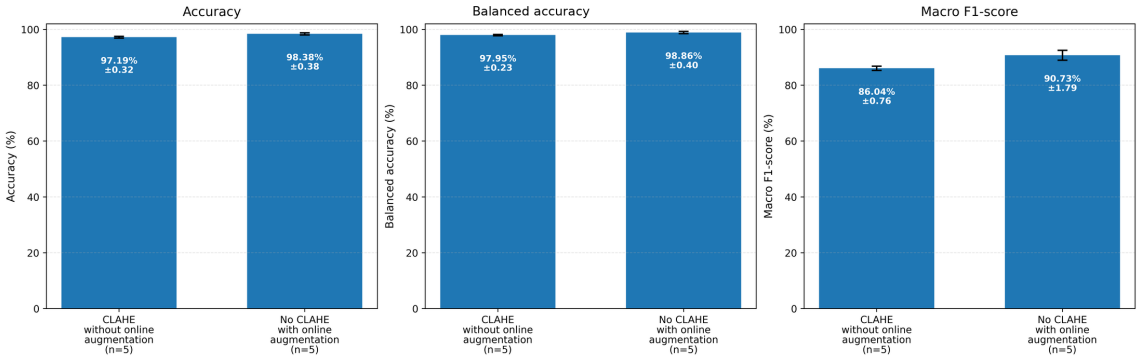


Fig. 5. Pipeline-screening comparison under the same split, model, 0.5/0.5 normalization, and five training seeds. The no-CLAHE configuration with online augmentation was the stronger of the two evaluated pipelines.

TABLE IV. SEED-LEVEL HELD-OUT RESULTS FOR THE SELECTED NO-CLAHE/AUGMENTATION CONFIGURATION. EACH ROW USES THE IDENTICAL FIXED IMAGE-LEVEL SPLIT

Seed	Best epoch	Accuracy (%)	Balanced accuracy (%)	Macro F1-score (%)	Macro-AUC
42	25	98.02	98.23	87.98	0.9994
52	25	98.96	99.32	91.91	0.9998
62	25	98.23	98.94	90.76	0.9996
72	25	98.13	98.77	92.66	0.9976
82	24	98.54	99.03	90.32	0.9999
Mean ± SD	–	98.38 ± 0.38	98.86 ± 0.40	90.73 ± 1.79	0.9993 ± 0.0009

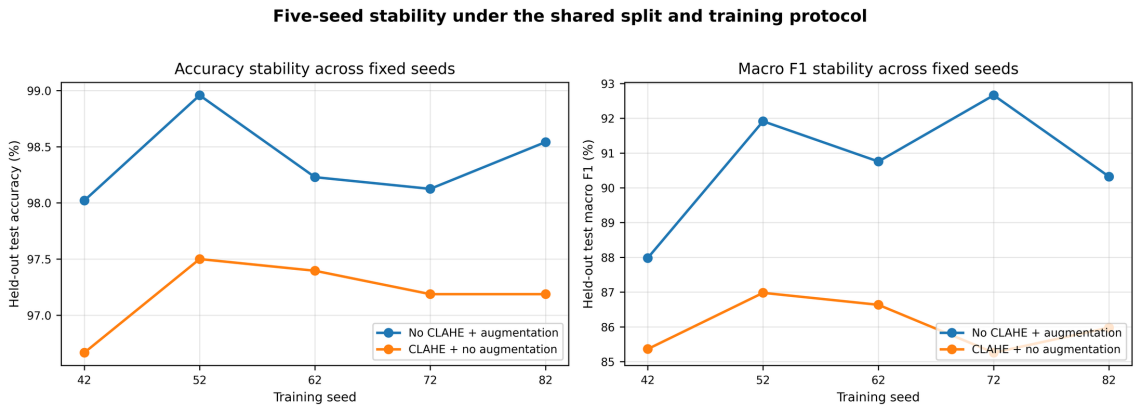


Fig. 6. Accuracy and macro F1-score across five fixed seeds. The values quantify seed-to-seed training variation under one shared image-level partition.

TABLE V. PER-CLASS RESULTS FOR THE SELECTED NO-CLAHE/AUGMENTATION CONFIGURATION. VALUES ARE MEAN ± STANDARD DEVIATION OVER FIVE SEEDS; SUPPORT IS THE FIXED TEST-SET COUNT

Class	Precision (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Support
Non-Demented	99.46 ± 0.68	98.58 ± 0.56	99.46 ± 0.68	99.02 ± 0.33	480
Very Mild Demented	99.28 ± 0.54	97.74 ± 0.98	99.62 ± 0.29	98.50 ± 0.51	336
Mild Demented	99.12 ± 0.80	99.11 ± 1.61	99.85 ± 0.13	99.11 ± 0.69	135
Moderate Demented	49.89 ± 7.64	100.00 ± 0.00	99.01 ± 0.31	66.29 ± 6.88	9



Fig. 7. Class-wise visualization based on five-seed aggregate results. Left: mean F1-score with standard deviation for both configurations. Right: selected configuration precision and sensitivity by class. The 100% Moderate Demented sensitivity is shown together with the nine-image fixed test support to provide the appropriate class-level context.

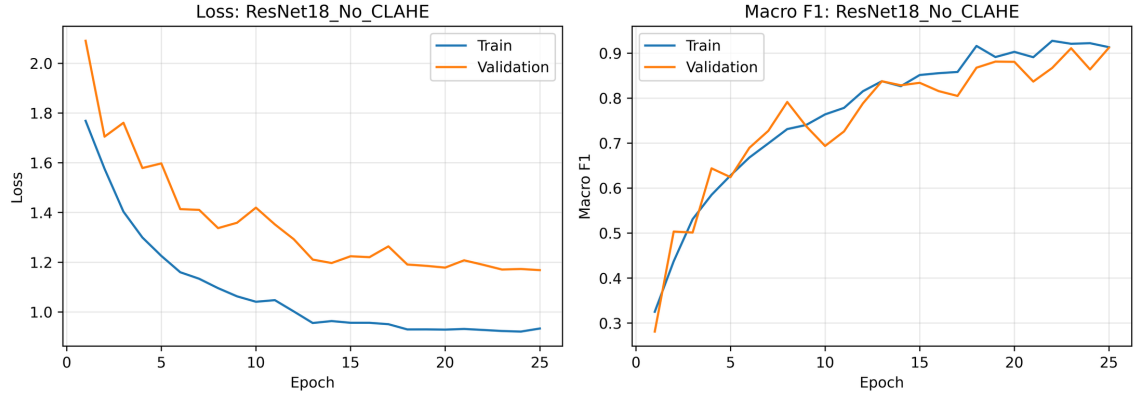


Fig. 8. Training and validation dynamics for the representative seed-42 selected run. The held-out test result was calculated only after restoring the lowest-validation-loss checkpoint.

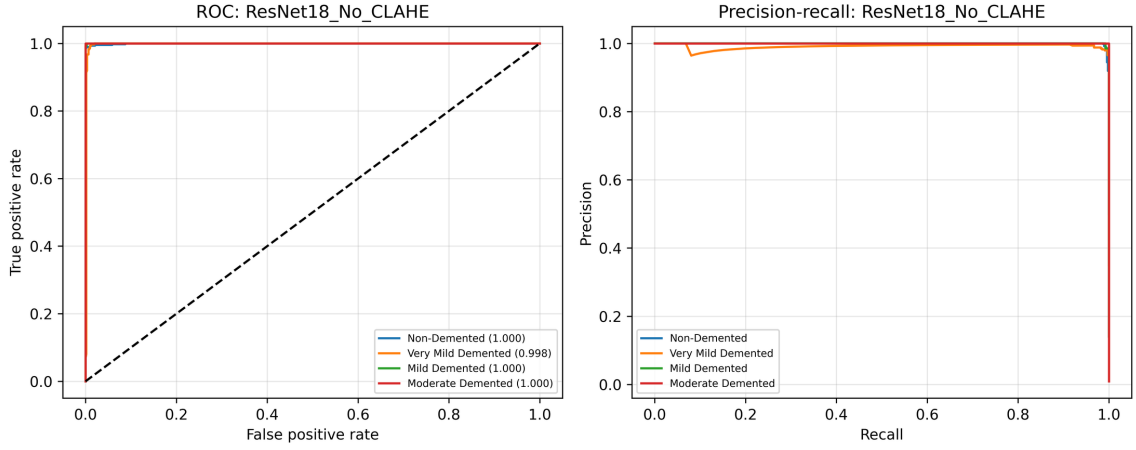


Fig. 9. One-versus-rest ROC and precision–recall curves for the representative seed-42 selected configuration on the held-out test set.

E. Confusion-Matrix Analysis

Fig. 10 provides absolute and row-normalized matrices for the representative seed-42 run. It complements the five-seed class-wise summary and makes the relationship between recall, false positives, and the nine-image Moderate Demented support directly visible. For this representative run, all 9 Moderate Demented test images were correctly identified,

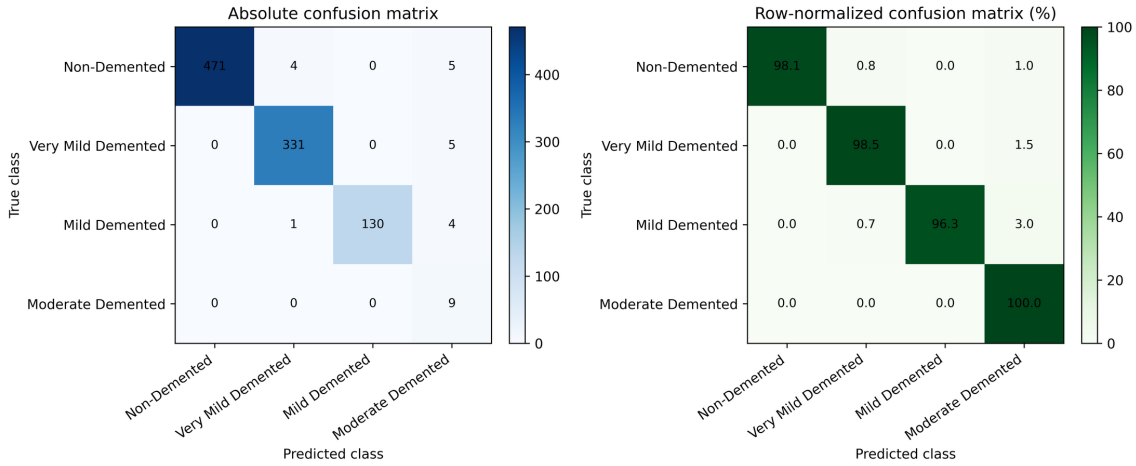


Fig. 10. Absolute and row-normalized confusion matrices for the representative seed-42 selected run. Interpret the figure together with the five-seed per-class summary in Table V.

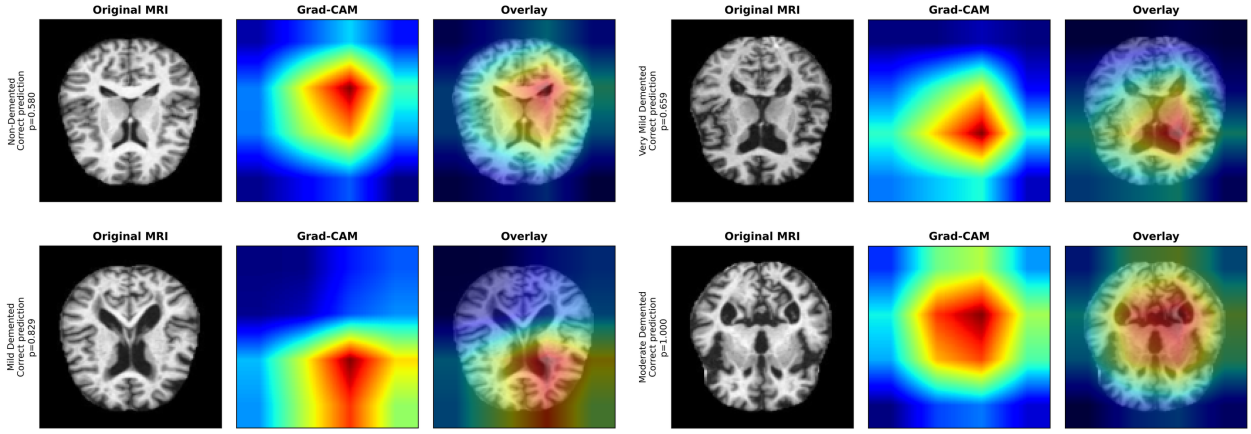


Fig. 11. Grad-CAM visualizations from the representative seed-42 checkpoint for correctly classified held-out test images from the four AD MRI classes. Each class panel shows the original grayscale MRI image, the Grad-CAM map, and the corresponding overlay. The displayed example for each class is the correctly classified image with confidence closest to the median confidence of correct predictions for that class. Warmer map values indicate locations with greater contribution to the predicted class. The maps are presented as qualitative model-inspection evidence, with quantitative anatomical validation reserved for future expert-annotated studies.

TABLE VI. CONFIDENCE CALIBRATION ON THE FIXED TEST PARTITION. TEMPERATURE WAS FITTED USING ONLY THE VALIDATION PARTITION

Output	NLL	Brier	ECE	MCE
Uncalibrated	0.4766	0.2613	0.3472	0.4188
Temperature-scaled	0.1756	0.0449	0.0722	0.5301

TABLE VII. SYNTHETIC ROBUSTNESS EVALUATION USING THE UNCHANGED SEED-42 CHECKPOINT. CHANGE IS RELATIVE TO CLEAN-TEST MACRO F1-SCORE

Condition	Accuracy (%)	Balanced accuracy (%)	Macro F1-score (%)	Δ Macro F1-score (pp)
Clean	98.02	98.23	87.98	0.00
Gaussian noise ($\sigma = 0.03$)	97.19	98.24	86.22	-1.76
Gaussian blur (radius 1.0)	93.65	96.15	79.42	-8.57
Brightness 0.80	98.02	98.65	88.62	$+0.63$
Brightness 1.20	95.10	95.71	82.24	-5.74
Contrast 0.80	98.02	98.52	89.48	$+1.49$
Contrast 1.20	96.56	97.68	84.66	-3.32
Rotation -5°	98.33	99.12	90.97	$+2.99$
Rotation $+5^\circ$	98.33	98.92	90.88	$+2.89$

TABLE VIII. MEASURED COMPUTATIONAL EFFICIENCY PROFILE

Property	Value
Trainable parameters	11,309,380
Checkpoint size	43.22 MB
GMACs per image	0.596
Estimated GFLOPs per image	1.191
CPU latency, batch 1	76.70 ms/image
GPU latency, batch 1	3.08 ms/image
GPU throughput, batch 32	3341 images/s
Peak GPU memory	126.88 MB

while 14 images from the other three classes were assigned to the Moderate Demented label, giving 9 true positives, 0 false negatives, and 14 false positives for that class.

F. Grad-CAM Explainability Results

Fig. 11 presents Grad-CAM visualizations for representative correctly classified images from the four classes. The seed-42 checkpoint used for this analysis obtained 98.02% accuracy, 98.23% balanced accuracy, and 87.98% macro F1-score on the fixed held-out test partition, consistent with the corresponding row in Table IV. The maps show the spatial distribution of image regions contributing to each predicted class. They are presented as qualitative model-inspection evidence; no anatomical localization claim is made because no expert-defined region-of-interest reference is used in this analysis.

G. Confidence Calibration and Selective Prediction

The validation-fitted temperature was $T = 0.2135$. As shown in Table VI, temperature scaling reduced NLL from 0.4766 to 0.1756, Brier score from 0.2613 to 0.0449, and ECE from 0.3472 to 0.0722. Accuracy and macro F1-score remained 98.02% and 87.98%, respectively, because scalar temperature scaling did not change class predictions. MCE increased from 0.4188 to 0.5301, indicating that a sparsely populated confidence bin retained a larger local calibration gap even though NLL, Brier score, and ECE improved. Fig. 12 therefore shows that the aggregate and maximum-bin calibration measures provide complementary information.

The confidence-ranked risk–coverage analysis is shown in Fig. 13. At 95% coverage, retained accuracy was 99.56% compared with 98.02% at full coverage. The temperature-scaled and uncalibrated curves were nearly identical because scalar temperature scaling preserves the score ordering apart from numerical ties. This experiment characterizes the confidence–coverage trade-off within the fixed test partition and provides a complementary view of confidence-ranked prediction behaviour.

H. Robustness Under Controlled Image Perturbations

Table VII and Fig. 14 summarize the synthetic stress tests. Gaussian noise produced a 1.76-percentage-point reduction in macro F1-score, whereas Gaussian blur produced the largest degradation, reducing macro F1-score from 87.98% to 79.42%.

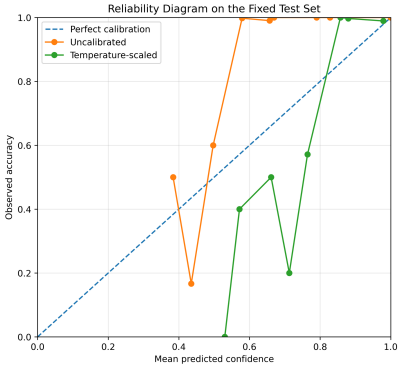


Fig. 12. Reliability diagram for the representative seed-42 checkpoint before and after validation-fitted temperature scaling.

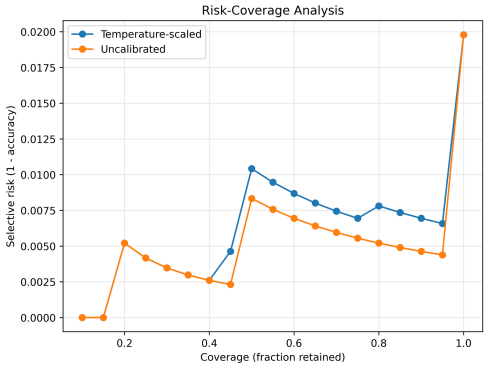


Fig. 13. Selective-prediction risk for the representative seed-42 checkpoint as a function of retained coverage.

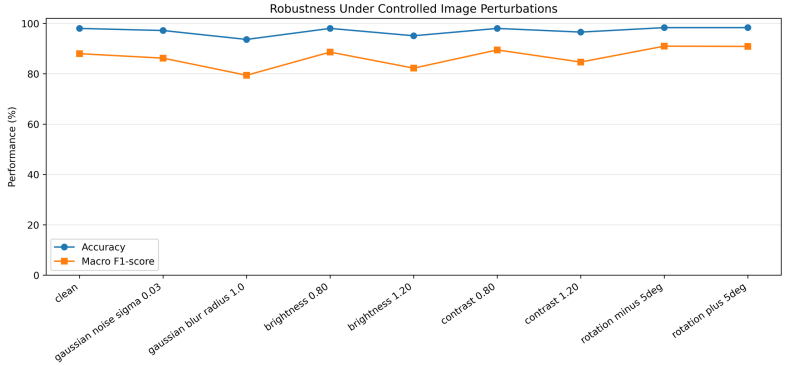


Fig. 14. Accuracy and macro F1-score for the representative seed-42 checkpoint under the specified synthetic test-time perturbations.

Increased brightness and contrast also reduced macro F1-score by 5.74 and 3.32 percentage points, respectively. Mild brightness reduction, contrast reduction, and $\pm 5^{\circ}$ rotations produced small apparent improvements on the fixed test partition. Because the Moderate Demented test support contains nine images, these increases are reported as fixed-sample variation and are not interpreted as a beneficial effect of perturbation.

I. Computational Efficiency Profile

The model contained 11,309,380 trainable parameters, occupied 43.22 MB as a checkpoint, and required 0.596 GMACs (approximately 1.191 GFLOPs) per 128×128 image. On the measured environment, CPU batch-1 inference averaged 76.70 ms per image. Tesla T4 batch-1 inference averaged 3.08 ms per image, while batch-32 throughput reached 3341 images/s with 126.88 MB peak allocated GPU memory. Table VIII provides the complete profile. These measurements support the resource-efficient characterization of the framework for the stated hardware, batch sizes, and runtime conditions.

V. Discussion

A. Main Finding

Under the evaluated protocol, the no-CLAHE/augmentation configuration provided the stronger aggregate performance of the two screened pipelines. It obtained a macro F1-score above 90% while remaining compact relative to many 3D and attention-heavy neuroimaging systems..

The comparison supports empirical evaluation of preprocessing choices within the target collection. CLAHE was included as a local-contrast candidate based on established adaptive-histogram principles [10], [11]. Under the evaluated protocol, the CLAHE-based pipeline without online augmentation produced lower aggregate accuracy, macro precision, and macro F1-score. Because the pipelines differ in both contrast enhancement and augmentation, the difference may reflect preservation of intensity structure, online augmentation, their interaction, or another configuration-level effect. A complete 2×2 factorial design remains a useful extension for estimating independent main effects and interactions.

B. Interpretation Under Imbalance

The difference between the high overall accuracy and the macro-level metrics is mainly associated with the Moderate Demented class. The selected configuration achieved 100% mean sensitivity for that class but lower precision, implying that some images from other classes were assigned to the rare label. This is why aggregate accuracy must not be used alone in imbalanced settings [9], [18]. Macro F1-score, macro precision, per-class support, and confusion patterns provide a more complete assessment.

The Moderate Demented class contains nine test images. Consequently, individual prediction changes have a visible effect on sensitivity and precision; for example, one additional false negative would change sensitivity by more than 11 percentage points. Reporting support together with precision, sensitivity, and F1-score therefore provides the most informative interpretation of this class.

C. Qualitative Explainability

The Grad-CAM maps provide a visual check that the representative seed-42 classifier drew evidence from spatially distributed regions within the MRI field rather than exclusively from image borders or uniform background. This observation is useful for model inspection. The heatmaps are intentionally presented as qualitative visual evidence because the analysis does not use expert segmentation, an anatomical atlas, or pathology-grounded region annotations. Consequently, Grad-CAM complements the predictive metrics without making an anatomical localization claim.

D. Reliability, Robustness, and Resource Efficiency

The calibration results show that high discrimination and reliable confidence are separate properties. For the seed-42 checkpoint, validation-fitted temperature scaling improved NLL, Brier score, and ECE without altering the predicted labels. MCE moved in the opposite direction, showing that aggregate calibration measures and the largest individual-bin deviation capture different aspects of probability behaviour. Selective prediction further showed that higher-confidence subsets had lower error; the full multiclass metrics remain the primary performance summary.

The controlled perturbation results provide a differentiated sensitivity profile. The framework remained comparatively stable under mild Gaussian noise and small rotations, while blur and intensity increases produced larger metric changes. These results complement clean-test performance, while scanner differences, reconstruction pipelines, motion effects, and institutional variation remain important directions for future multi-centre evaluation.

The measured 0.596 GMACs per image, 43.22 MB checkpoint, and low Tesla T4 inference latency provide direct evidence for resource efficiency. Future translational evaluation can extend this evidence through external validation, use-case-specific calibrated thresholds, software-system testing, and prospective workflow assessment.

E. Study Scope and Future Work

The scope of the present study is a reproducible image-level evaluation on one public MRI collection. The available data support image-level rather than verified subject-level partitioning, and the reported five-seed standard deviations therefore summarize training reproducibility under one fixed split. The Moderate Demented class is reported with its nine-image test support so that class-specific metrics are interpreted with their sample context. The framework uses 2D images, while volumetric and multi-slice representations offer a natural extension when complete acquisition context is available. Grad-CAM is used for qualitative model inspection, calibration is assessed within the supplied dataset partitions, and the perturbation study is restricted to the specified synthetic transformations. Computational measurements are reported for the stated hardware and runtime configuration.

Future work can extend the present framework through datasets with subject-level and site-level identifiers, external multi-centre evaluation, volumetric or multi-slice representations with preserved acquisition context. Additional directions

include calibration assessment under dataset shift, use-case-specific selective-decision thresholds, quantitative explainability assessment against expert-defined anatomical references, and a complete 2×2 factorial study of CLAHE and augmentation.

VI. Conclusion

This study developed a reliability-oriented and resource-efficient evaluation of a lightweight ResNet-18 framework for four-class AD MRI image classification. Under a fixed image-level split and five training seeds, the selected no-CLAHE configuration with online augmentation achieved 98.38% $\pm$ 0.38% accuracy, 98.86% $\pm$ 0.40% balanced accuracy, 90.73% $\pm$ 1.79% macro F1-score, and 0.9993 $\pm$ 0.0009 macro-AUC. For the representative seed-42 checkpoint, temperature scaling improved NLL, Brier score, and ECE, while selective prediction characterized the confidence–coverage trade-off. Controlled stress tests showed larger performance changes under Gaussian blur and intensity increases than under mild noise or small rotations. The model required 0.596 GMACs per image and achieved 3.08 ms batch-1 inference on a Tesla T4, providing measured evidence of resource efficiency. Grad-CAM provided qualitative model-inspection evidence without making an anatomical localization claim. Collectively, the findings support a reproducible, confidence-aware, perturbation-tested, and computationally profiled image-level framework within the evaluated public dataset and protocol.

References

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, C. I. Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88. doi:10.1016/j.media.2017.07.005.

[2] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (2017) 221–248. doi:10.1146/annurev-bioeng-071516-044442.

[3] J. Wen, E. Thibeau-Sutre, M. Diaz-Melo, J. Samper-González, A. Routier, S. Bottani, D. Dormont, S. Durrleman, N. Burgos, O. Colliot, Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation, Med. Image Anal. 63 (2020) 101694. doi:10.1016/j.media.2020.101694.

[4] M. A. Ebrahimighahnavieh, S. Luo, R. Chiong, Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review, Comput. Methods Programs Biomed. 187 (2020) 105242. doi:10.1016/j.cmpb.2019.105242.

[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[6] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.

[7] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, J. Liang, Convolutional neural networks for medical image analysis: Full training or fine tuning?, IEEE Trans. Med. Imaging 35 (5) (2016) 1299–1312. doi:10.1109/TMI.2016.2535302.

[8] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 60. doi:10.1186/s40537-019-0197-0.

[9] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Netw. 106 (2018) 249–259. doi:10.1016/j.neunet.2018.07.011.

[10] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. M. ter Haar Romeny, J. B. Zimmerman, K. J. Zuiderveld, Adaptive histogram equalization and its variations, Comput. Vision Graph. Image Process. 39 (3) (1987) 355–368. doi:10.1016/S0734-189X(87)80186-X.

[11] K. J. Zuiderveld, Contrast limited adaptive histogram equalization, in: P. S. Heckbert (Ed.), Graphics Gems IV, Academic Press Professional, San Diego, CA, USA, 1994, pp. 474–485. doi:10.1016/B978-0-12-336156-1.50061-6.

[12] J. S. Kim, J. W. Han, J. B. Bae, et al., Deep learning-based diagnosis of Alzheimer's disease using brain magnetic resonance images: An empirical study, Sci. Rep. 12 (2022) 18007. doi:10.1038/s41598-022-22917-3.

[13] L. Chen, H. Qiao, F. Zhu, Alzheimer's disease diagnosis with brain structural MRI using multiview-slice attention and 3D convolution neural network, Front. Aging Neurosci. 14 (2022) 871706. doi:10.3389/fnagi.2022.871706.

[14] F. Hajamohideen, et al., Four-way classification of Alzheimer's disease using deep Siamese convolutional neural network with triplet-loss function, Brain Inform. 10 (2023) 5. doi:10.1186/s40708-023-00184-w.

[15] P. Carcagnì, M. Leo, M. Del Coco, C. Distante, A. De Salve, Convolution neural networks and self-attention learners for Alzheimer dementia diagnosis from brain MRI, Sensors 23 (3) (2023) 1694. doi:10.3390/s23031694.

[16] J. Xin, A. Wang, R. Guo, W. Liu, X. Tang, CNN and Swin-Transformer based efficient model for Alzheimer's disease diagnosis with sMRI, Biomed. Signal Process. Control 86 (2023) 105189. doi:10.1016/j.bspc.2023.105189.

[17] B. S. Rao, M. Aparna, S. S. Kolisetty, H. Janapana, Y. V. Koteswararao, Multi-class classification of Alzheimer's disease using deep learning and transfer learning on 3D MRI images, Traitement du Signal 41 (3) (2024) 1397–1404. doi:10.18280/ts.410328.

[18] T. Saito, M. Rehmsmeier, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, PLOS ONE 10 (3) (2015) e0118432. doi:10.1371/journal.pone.0118432.

[19] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874. doi:10.1016/j.patrec.2005.10.010.

[20] A. Yakkundi, Alzheimer's disease dataset, Mendeley Data (2023). doi:10.17632/ch87yswbz4.1.

[21] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 448–456.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

[23] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2019.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, D. Wojna, Rethinking the Inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826. doi:10.1109/CVPR.2016.308.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626. doi:10.1109/ICCV.2017.74.

[26] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: Proceedings of the 34th International Conference on Machine Learning, PMLR 70, 2017, pp. 1321–1330.

[27] Y. Geifman, R. El-Yaniv, Selective classification for deep neural networks, in: Advances in Neural Information Processing Systems 30, 2017.

[28] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: International Conference on Learning Representations, 2019.

Multi-Level Contextual Semantic Modeling for Implicit Hate Speech Recognition

Rajkumar Panchal[1]*, Megha Jonnalagedda[2]
Department of Computer Science and Engineering,
SGGSIE&T, Swami Ramanand Teerth Marathwada University, Nanded, 431606, Maharashtra, INDIA[1]
Department of Information Technology,
SGGSIE&T, Swami Ramanand Teerth Marathwada University, Nanded, 431606, Maharashtra, INDIA[2]

***Abstract*—Social media platforms are key spaces for sharing opinions, experiences, and daily events. However, the rapid rise in hate speech, especially implicit hate, has become a significant concern. Implicit hate is subtle, context-dependent, and more difficult to detect than explicit hate. Despite the growing body of work on Hate Speech Detection (HSD), most existing models are developed for resource-rich languages like English. Unlike resource-rich languages, Hindi lacks sufficient annotated data and dedicated research, making it a low-resource language. To address this limitation, this work presents a deep learning–driven approach for the classification of implicit and explicit hate speech in Hindi. The approach uses FastText embeddings to capture semantic and subword-level information, which suits noisy social media text. A Bidirectional Long Short-Term Memory (BiLSTM) network models contextual dependencies, while an attention mechanism assigns higher importance to discriminative words and sentences, enabling effective identification of implicit hate. Extensive experiments on the Hindi Hate Speech Dataset (HHSD) demonstrate the model's effectiveness. Compared with the widely used multilingual BERT (mBERT) baseline, the proposed approach improves classification performance, with a 2.51% increase in accuracy and a 17.84% improvement in weighted F1-score. Additionally, the model requires significantly fewer parameters than transformer-based models, making it more parameter-efficient.**

***Keywords*—Attention network; BiLSTM; deep learning; explicit hate speech detection; hate speech detection; implicit hate speech detection; natural language processing**

I. Introduction

Social media has become an integral platform for sharing opinions, daily activities, and connecting with others. While many users engage positively, others disseminate negative content, including hate speech, which can undermine societal harmony. Hate speech encompasses communicative acts expressed through language or behaviour that demean or discriminate against individuals or communities based on attributes such as religion, ethnicity, nationality, race, colour, descent, gender, or other identity markers [1]. To address this issue, governments have introduced regulations that require social media platforms to enforce them. However, to bypass these restrictions, users increasingly resort to indirect or implicit forms of hate, conveyed through subtle textual expressions or multimodal content. Implicit hate speech is characterized by indirect or veiled expressions that convey negative or prejudicial attitudes toward individuals or groups defined by protected identity attributes [2]. Implicit hate speech avoids overtly abusive language and instead uses context and underlying meaning to convey hostility, which makes detection much more difficult. This complexity is further amplified by cultural and linguistic diversity, where expressions considered acceptable in one context may be offensive in another. In countries like India, where many languages are spoken, people often mix languages, and informal language substantially increases the difficulty of Hate Speech Detection [3], particularly for implicit hate [4]. The slow, tedious process of manual moderation has driven a shift toward automated HSD systems based on machine learning and deep learning techniques. However, most research focuses on English, a high-resource language, while Indian languages remain underexplored [3]. The scale of Hindi usage contrasts sharply with the limited research on implicit hate speech detection in the language, underscoring the need for effective automated solutions.

The main contributions of the proposed work are:

- Contextual modeling for Implicit Hate Detection: This work investigates the hierarchical integration of BiLSTM and attention to detect implicit and explicit hate speech in Hindi. The framework models contextual information at two levels: word-level representations capture local semantic dependencies, while sentence-level representations capture broader contextual relationships.
- Task-Specific Hierarchical Attention: The proposed framework employs attention at both the word and sentence levels to selectively aggregate contextual representations. The methodological contribution lies in using hierarchical attention to identify salient lexical and sentence-level contextual cues relevant to implicit versus explicit hateful intent. This provides a more targeted representation for fine-grained hate classification.
- Emoji-Preserving Contextual Processing for Hindi Social Media Text: Unlike preprocessing pipelines that discard emojis as non-linguistic symbols, the proposed framework preserves emojis as individual tokens because they can provide emotional and pragmatic cues that complement textual meaning. This design is particularly relevant to Hindi social-media content, where emojis may shape interpretations of indirect or context-dependent expressions.
- Semantic-Preserving GPT-3.5 Backtranslation Framework: We employ a GPT-3.5-based backtranslation

*Corresponding author.

strategy to address class imbalance and increase linguistic diversity. We validate augmented-data quality using LaBSE similarity and mBERTScore to ensure semantic consistency between original and generated samples.

- Integrated Low-Resource Hate Speech Detection Framework: The main methodological contribution is integrating hierarchical contextual modeling, dual-level attention, emoji preservation, and semantically validated augmentation into a single framework for Hindi implicit/explicit hate speech detection.

II. Related Works

A. Machine Learning Approaches

Initial research on hate speech detection primarily utilized machine learning (ML) methods that relied on handcrafted features, including n-grams, Bag-of-Words, sentiment cues, syntactic dependencies, and hate lexicons. Classical classifiers, including Decision Trees, Bayesian models, Random Forests, Logistic Regression and Support Vector Machines have been widely employed for hate speech detection in English [5], [6], [7]. Similar approaches were later applied to Hindi and code-mixed Hindi–English text using FastText at character-level representations and traditional classifiers [8]. Despite their initial success, ML-based methods suffer from several inherent limitations. These approaches depend on manual feature engineering and domain-specific lexicons, which increases errors, requires significant time, and limits scalability. Furthermore, they are biased toward explicit hate containing abusive words and fail to capture deeper contextual semantics, sarcasm, and implicit hate expressions. Their performance also degrades significantly in code-mixed and multilingual environments, which are common in Indian social media platforms. As a result, ML techniques are insufficient for robust implicit/explicit hate speech detection in Hindi.

B. Deep Learning Approaches

To overcome the limitations of ML-based models, researchers increasingly adopted deep learning (DL) architectures that can automatically extract features and improve contextual representation. Neural models such as hierarchical networks, BiLSTMs, LSTMs, CNNs, and attention-based architectures demonstrated enhanced semantic understanding and contextual modeling [9], [10], [11]. Recent work has explored multimodal toxicity detection in memes by combining PaddleOCR, KOSMOS-2, and SigLIP to analyse textual and visual cues jointly [12]. For Indian languages, several studies have incorporated subword representations, phrase embeddings, sentiment and emotion features, and auxiliary linguistic resources to handle linguistic diversity and morphological richness better.

Recent studies have increasingly adopted pretrained transformer-based models, such as mBERT, MuRIL, BERT, and HateBERT, word clouds, and ensemble transformer architectures for hate speech detection across diverse languages, including Hindi, Marathi, Tamil, and English [13], [14], [15], [16]. Although these models achieve strong empirical performance, they introduce critical challenges, including large parameter sizes, high computational costs, and dependency on GPUs/TPUs and large annotated datasets. Additionally, many transformer-based approaches exhibit bias toward majority classes, struggle with unknown tokens, sarcasm, and code-mixed text, and often fail to distinguish implicit hate from explicit hate effectively. These challenges highlight the need for parameter-efficient deep learning models better suited to low-resource language settings.

C. Implicit and Explicit Hate Speech Detection

Implicit hate speech detection has gained increased attention as it relies on indirect expression, nuanced meaning, and strong contextual dependence. Unlike explicit hate, implicit hate avoids overtly abusive language, making it difficult for lexicon-based and surface-level models to detect. Elsherief et.al. [4] introduced benchmark datasets and annotation guidelines that distinguish between implicit and explicit hate, laying the foundation for further research in this area. Subsequent studies explored background knowledge integration, contrastive learning-based fine-tuning strategies [17], prompt-based learning approaches, and large language models to improve interpretability and performance [18].

For Indian languages, research on implicit hate detection remains limited. Kapil et. al. [3] introduced the Hindi Hate Speech Dataset (HHSD) and evaluated implicit and explicit hate using transformer-based architectures.

Despite significant progress in hate speech detection, existing approaches for Hindi largely rely on traditional machine learning methods or computationally intensive pretrained transformer models, both of which struggle to handle implicit hate speech effectively. Furthermore, most studies focus solely on textual content and overlook emoji-based contextual cues, even though emojis are frequently used on social media to convey sentiment, sarcasm, and indirect intent. Robust experimental evaluation of implicit and explicit hate classification remains limited, with many studies relying on single train–test splits rather than comprehensive validation strategies. In addition, data imbalance and limited dataset size remain challenges, as augmentation techniques are rarely explored in the context of implicit hate detection. These limitations highlight the need for efficient, context-aware, and well-evaluated models tailored to the Hindi language (a low-resource language).

Existing Hindi hate speech detection methods primarily rely on transformer-based architectures or keyword-driven approaches that struggle to capture implicit hateful intent. Furthermore, limited attention has been given to multi-level contextual modeling, interpreting model focus, emoji-aware contextual understanding, and semantically validated augmentation strategies. To address these limitations, this work develops a task-specific hierarchical contextual framework that combines word- and sentence-level BiLSTM representations with dual-level attention for implicit/explicit hate discrimination. The framework additionally preserves emojis as contextual tokens and incorporates GPT-3.5-based back-translation with semantic validation using LaBSE and mBERTScore.

This study introduces a novel approach to address the challenges of identifying both implicit and explicit hate speech in Hindi. Emojis are widely used on social media to convey emotions and implicit intent; incorporating emoji information is essential for accurately interpreting user-generated content

[19]. Accordingly, we retain and process emojis during preprocessing to enhance contextual understanding. The proposed model uses FastText embeddings to generate efficient, semantically rich representations of the input text [20]. These embeddings flow into a Bidirectional Long Short-Term Memory network, which uncovers contextual relationships from both past and future perspectives. Furthermore, an attention mechanism assigns higher importance to salient words or sentences, enabling the model to focus on critical contextual cues for detecting implicit hate speech [21].

III. Proposed Methodology

The proposed technique classifies hate posts on social media as either implicit or explicit hate speech. Fig. 1 shows the system architecture. The proposed system takes Hindi text as input, which is then preprocessed. Using an embedding technique, the text is converted into a vector and, finally, classified as implicit/explicit hate speech. The detailed explanation is as follows.

A. Dataset

The proposed method uses the Hindi Hate Speech Dataset (HHSD). HHSD contains 14782 tweets, each labelled as hate or non-hate. 7311 hateful tweets are present, which are further labelled as implicit or explicit hate speech. Of these, 2432 are implicit, and 4879 are explicit. Because the proposed work focuses on binary classification, we applied the method only to tweets with hate labels. Table I presents detailed information about the dataset.

TABLE I. Label Distribution

Label	Label	Size
Hate	Implicit	2432
	Explicit	4879
Nonhate		7471
Total		14782

B. Motivation and Design Rationale

The proposed approach's key design consideration is a hierarchical contextual modeling framework that captures context at both the word and sentence levels. Unlike traditional hate speech detection models that rely primarily on offensive keywords, the framework identifies subtle and indirect forms of hate speech by focusing on contextual semantics and inter-sentence relationships. At the word level, attention is used to highlight critical terms that signal hateful intent, even when explicit abusive language is absent. Additionally, at the sentence level, attention identifies which sentences carry the most discriminatory meaning within multi-sentence posts. This hierarchical attention structure provides an interpretable view of the words and sentences that influence predictions, aiming to improve detection of challenging implicit hate speech scenarios. By combining bidirectional contextual learning with multi-level attention, the model provides complementary contextual representations of textual content compared with conventional single-level attention or keyword-based approaches.

The methodological contribution therefore lies in integrating these established components for task-specific contextual discrimination of implicit and explicit hate speech.

1) Context-aware implicit hate speech detection: Traditional hate speech detection models often rely on the presence of explicit offensive terms (e.g., slurs, abusive words, derogatory expressions). However, implicit hate speech is expressed indirectly through stereotypes, sarcasm, coded language, insinuations, or contextual references. Since no explicit hate-related keywords may be present, detecting implicit hate requires understanding the surrounding linguistic context.

For example:

- Explicit Hate: "आप जैसे हरामी को भी अवसर मिलेगा" (Translation: Even a scoundrel like you will get an opportunity.)
- Implicit Hate:"मानसिक दिवालियापन है 🤬🤬" (Translation:It is mental bankruptcy 🤬🤬)

The second statement contains no obvious hate words, but the context implies discrimination toward a particular group. Therefore, the proposed model utilizes contextual representation learning through BiLSTM and attention mechanisms.

The meaning of a potentially hateful expression depends heavily on its surrounding context. A word or phrase may appear neutral on its own but may express discriminatory intent when interpreted in a broader linguistic context. Therefore, contextual representation learning is essential for accurately distinguishing implicit hate speech from explicit hate speech.

Let the input sentence be represented as in Eq. (1):

$$S = \{w_1, w_2, w_3, \ldots, w_l\} \tag{1}$$

where, w_i denotes the ith word and l represents the sentence length. Each word is transformed into a dense vector representation using Eq. (2):

$$x_i = Embedding(w_i) \tag{2}$$

A Bidirectional Long Short-Term Memory (BiLSTM) network processes the embedded sequence. The forward hidden state is computed using Eq. (3):

$$\overrightarrow{h_i} = LSTM_f(x_i, \overrightarrow{h_{i-1}}) \tag{3}$$

and the backward hidden state is computed as defined in Eq. (4):

$$\overleftarrow{h_i} = LSTM_b(x_i, \overleftarrow{h_{i+1}}) \tag{4}$$

The contextual representation of each word is obtained Eq. (5) by concatenating the forward and backward hidden states :

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{5}$$

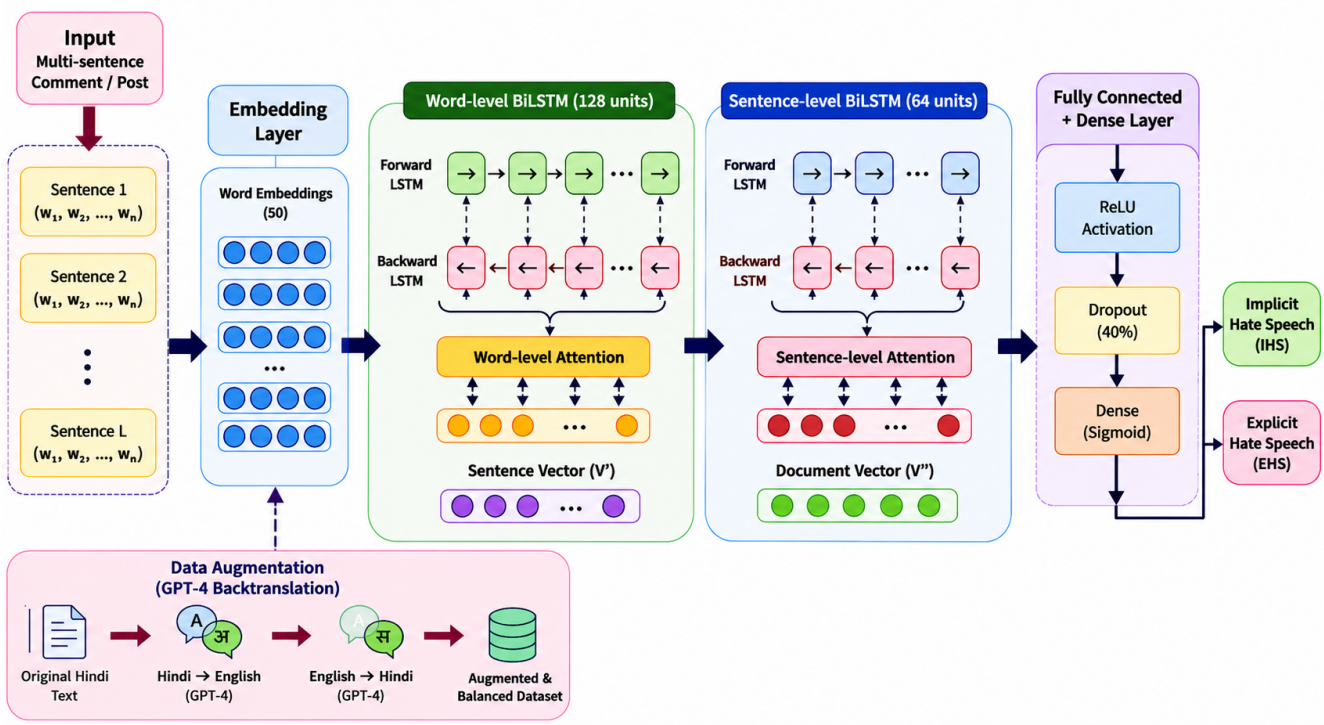


Fig. 1. Architecture diagram.

where, $h_i \in \mathbb{R}^{2d}$ and d denotes the hidden-state dimension. The BiLSTM output matrix is represented using Eq. (6):

$$0B^{(w)}_{BiLSTM} = [h_1, h_2, \ldots, h_l] \in \mathbb{R}^{l \times 2d} \tag{6}$$

Unlike explicit hate speech, implicit hate often lacks direct offensive words. Therefore, contextual understanding becomes essential. The BiLSTM captures semantic relationships among words, enabling the model to identify hidden hateful intent even when explicit abusive vocabulary is absent.

2) Hierarchical multi-level context modeling: Hateful intent may span multiple sentences rather than being expressed within a single sentence. Consequently, effective hate speech detection requires modeling both local semantic relationships among words and global contextual relationships among sentences.

At the word level, contextual dependencies among tokens are captured using a BiLSTM Eq. (7):

$$H_w = BiLSTM(EH) \tag{7}$$

where, EH denotes the FastText embedding representation. The resulting sentence representations are then processed through a sentence-level BiLSTM Eq. (8):

$$H_s = BiLSTM(V) \tag{8}$$

where, V represents the sentence vectors generated from the word-level attention mechanism. This hierarchical architecture enables the model to capture:

- Local semantic dependencies among words.
- Global discourse-level relationships among sentences.
- Contextual information distributed across multi-sentence posts.

By jointly modeling lexical and discourse-level context, the framework provides richer semantic representations for implicit hate speech detection.

3) Attention-guided feature selection: Not all words and sentences contribute equally to hate speech classification. In implicit hate speech, only a small subset of words may reveal the underlying hateful intent. Similarly, within a multi-sentence post, only certain sentences may contain discriminatory cues.

Overall, the proposed framework combines context-aware representation learning, hierarchical multi-level context modeling, and attention-guided feature selection to capture subtle and context-dependent hate expressions. Furthermore, we incorporate semantic-preserving back-translation to address data imbalance and increase linguistic diversity, supporting robust hate speech detection in the low-resource Hindi setting.

C. Preprocessing

During preprocessing, we cleaned the data. We removed content that does not contribute to classification, such as usernames, mentions, handles, hashtags, and hyperlinks. We removed leading, trailing, and multiple spaces from the input data. Special characters do not contribute to classification, so we also removed them from the input. Suffixes are not useful in classification tasks; therefore, we also removed them from the input text. To retain contextual knowledge, some stop

words play an important role, so preprocessing didn't remove them. FastText embedding converts the input into a vector representation using contextual knowledge, so keeping some stop words is important.

Many users across platforms, such as social media, use emojis to express their feelings in posts. Emojis provide a more expressive and attractive way to convey feelings or add meaning to textual posts. Emojis may be used to remove ambiguity from ambiguous sentences. Emojis are used to complement relatively negative text [19]. Emojis greatly contribute to understanding emotion, whether positive/negative and direct/indirect. Many researchers have considered emotion/sentiment as an important feature in HSD, specifically in implicit HSD [22], [23], [24].

Emojis benefit implicit/explicit HSD, so the proposed method does not remove them from the input data. Preprocessing uses them to understand the sentence's context better. In tokenization, each unique emoji is assigned a separate token, which is then used in embedding and other layers.

D. Tokenizer

The tokenizer converts input text into numerical representations, enabling more efficient processing by the NLP model. The tokenizer takes text and emojis as input, identifies unique words (w_i) and emojis, and builds a vocabulary. The vocabulary contains words, emojis, and their corresponding index entries, which are used to convert input into numbers. If any out-of-vocabulary words are present, they are handled using an out-of-vocabulary (OOV) token. In the proposed method, we replace OOV with the <UNK> token.

The tokenizer converts each word and emoji into a token t_i. This creates a numeric vector t of varying length, depending on the input sentence. The input vector t is padded to produce a fixed-length vector P of length l, $|P| = l \geq |t|$. The first $|t|$ entries are filled with t, and the remaining $(l - |t|)$ entries are filled with zeros as padding. In the proposed model, the maximum sentence length (l) is 50, so padding converts all input vectors to size 50. The resulting vector with fixed length $P \in \mathbb{R}^{1\times l}$ is provided to the next layer (embedding layer) for further processing.

E. Embedding Layer

Embedding converts input into an efficient representation, so this work uses FastText embeddings. Facebook provides pre-trained FastText embeddings [20] for 157 languages, including Hindi. FastText embeddings provide more efficient word representations. FastText provides an efficient data representation, which also makes the classification task easier. FastText captures word morphology and handles out-of-vocabulary words better. FastText extends word2vec, in which words are represented as a collection of subword embeddings. Because of these features, FastText is widely used in NLP tasks such as sentiment analysis and classification. Each token t_i of the input vector is embedded using FastText. This embedding creates a vector $E_H \in \mathbb{R}^{l\times 300}$. The embedding vector is 50x300. This vector proceeds to the next layer, the BiLSTM layer. It handles word morphology and out-of-vocabulary words better.

F. BiLSTM Layer

A bidirectional LSTM (BiLSTM) processes information in both directions. In the forward direction, an LSTM processes the input from left to right, and in the backward direction, from right to left. Due to this bidirectionality, the model captures context from the previous and next words relative to the current word. This provides a better representation of each word by considering the context of nearby words. So, the proposed method uses a BiLSTM to process the input in both directions and capture context from nearby words.

The embedding layer produces a vector that is fed into a BiLSTM. In implicit/explicit hate speech, knowing contextual information is important. Specific word-based identification is not available for detecting implicit hate. The implicit nature of the input makes the detection task much harder, so understanding context and representing it efficiently is important. In the proposed model, we use 128 BiLSTM units. BiLSTM provides b_i hidden state for each token t_i. By processing all tokens, we create a matrix using Eq. (9).

$$B_{BiLSTM} \in \mathbb{R}^{l\times 2d} \tag{9}$$

Where, d is the dimension of the hidden state.

l: length of the input vector (50).

This vector representation is then forwarded to the attention layer for further processing.

G. Attention Layer

The model learns word-specific importance weights through a lexical-level attention mechanism. This process assigns different importance scores to different words. Assigning importance scores allows the model to prioritize the most relevant words within sentences [21]. The resulting attention weights show the relative importance assigned to different tokens during classification.

For implicit/explicit HSD, the focus should be on the words. In explicit hate speech, the use of offensive/abusive words directly represents hate, but in implicit hate, there is no direct word present. Indirect targeting of individuals or groups complicates the identification of implicit hate speech, as it does not rely on explicit offensive or abusive language. So, we need to focus on the words in sentences that signal implicit hate speech. Attention mechanisms focus on words/sentences to identify important terms that signal implicit hate speech. The attention network accepts the B_{BiLSTM} matrix as input and assigns relative importance to tokens by focusing on words that contribute to implicit/explicit hate classification.

To automatically identify informative textual units, we apply an attention mechanism over the contextual representations using Eq. (10). First, a hidden representation is computed as:

$$U^{(w)} = tanh(B^{(w)}_{BiLSTM} W^{(w)} + b^{(w)}) \tag{10}$$

where, $B^{(w)}_{BiLSTM}$ represents the contextual word representations generated by the word-level BiLSTM, $W^{(w)}$ is a trainable weight matrix, and $b^{(w)}$ is the bias vector. The

importance score for each word is calculated as shown in Eq. (11):

$$s_i^{(w)} = U_i^{(w)} u^{(w)} \tag{11}$$

where, $u^{(w)}$ is the trainable word-level attention score vector. Attention weights are obtained using the softmax function, as defined in Eq. (12):

$$\alpha_i^{(w)} = \frac{\exp(s_i^{(w)})}{\sum_{j=1}^{l} \exp(s_j^{(w)})} \tag{12}$$

such that:

$$\sum_{i=1}^{l} \alpha_i^{(w)} = 1 \tag{13}$$

The context-aware sentence representation is then computed using Eq. (14):

$$V^{(s)} = \sum_{i=1}^{l} \alpha_i^{(w)} h_i \tag{14}$$

The word-level attention mechanism assigns different weights to contextual word representations, allowing the model to emphasize informative words within each sentence. These attention weights provide an interpretable view of the model's focus, but they should not be regarded as a complete explanation of the model's decision.

H. Sentence-Level BiLSTM Layer

The representations produced by the preceding layer are propagated in both forward and backward directions, enabling the model to capture complex inter-sentential relationships. This sentence-level BiLSTM provides contextual sentence representations by leveraging surrounding sentences in multi-sentence comments/posts. In the proposed model, we use 64 BiLSTM units.

I. Sentence-Level Attention Layer

The BiLSTM outputs feed into the sentence-level attention layer. This attention layer produces a single vector from multiple sentences, assigning higher weight to important sentences. This layer identifies relationships among sentences and focuses on the most important ones to form a single multi-sentence vector. Using this weighted vector, identifying implicit hate speech becomes easier.

The attention network creates a hidden representation using the tanh(.) activation function, as defined in Eq. (15).

$$U^{(s)} = \tanh(B_{BiLSTM}^{(s)} \cdot W^{(s)} + b^{(s)}) \tag{15}$$

Where, $w^{(s)}$ and $b^{(s)}$ are the trainable sentence-level attention parameters.

$B_{BiLSTM}^{(s)}$: Input matrix of size $\mathbb{R}^{l \times 2d}$.

Using Eq. (16), we compute the sentence potential importance score.

$$S_j^{(s)} = U_j^{(s)} u^{(s)} \tag{16}$$

Where, $u^{(s)}$ is the trainable sentence-level attention score vector.

Eq. (17) calculates attention weights using softmax(.) function.

$$\alpha_j^{(s)} = \frac{\exp\left(s_j^{(s)}\right)}{\sum_{k=1}^{m} \exp\left(s_k^{(s)}\right)} \tag{17}$$

Eq. (18) calculates the context vector as the sum of the products of the attention network input and the sentence-level attention weights.

$$V^{(d)} = \sum_{j=1}^{m} \alpha_j^{(s)} h_j^{(s)} \tag{18}$$

The sentence-level attention mechanism assigns different weights to contextual sentence representations, allowing the model to emphasise informative sentences among multiple sentences. The outcome of this layer is a sentence vector, which will be used for further processing in the model.

J. Fully Connected and Dense Layer

After attention-based feature extraction, a fully connected layer with ReLU non-linearity performs classification, with regularization via a 40% dropout strategy. Final predictions are generated using a softmax-based decision layer to distinguish between implicit hate speech (IHS) and explicit hate speech (EHS).

K. Model with Augmentation

To examine the impact of data augmentation and class balancing, we apply back-translation to the entire dataset. The size of the dataset after complete dataset augmentation and class balancing is shown below in Table II.

TABLE II. AUGMENTED DATASET DETAILS

Detail	Size	
	Implicit	Explicit
Class Balancing	4864	4879
Complete Augmentation	4864	9651

The HHSD dataset is highly imbalanced. As shown in Table I, the dataset contains 2432 tweets with implicit labels and 4879 with explicit labels. Using back-translation, a data augmentation technique, we balanced the dataset. In backtranslation, we first translate the input data from Hindi to English and then back to Hindi. For back-translation, we use GPT-3.5. Fig. 2 shows the back-translation process.



Fig. 2. Back-translation.

1) GPT-based back-translation: To augment the Hindi dataset, we performed back-translation using the OpenAI API with the `gpt-3.5-turbo` model. We first translated each selected Hindi instance into English and then translated it back into Hindi. This process aimed to generate semantically preserving variations of the original instances.

The translation was performed using the following prompt template:

> *Translate the following text to [target language]: [text]*

We also provided a system instruction identifying the model as a professional translator. We used the API's default settings. We used the resulting Hindi sentences as augmented instances while preserving the original class labels. To prevent possible data leakage during augmentation, we generated back-translated samples only within the training portion of each cross-validation fold.

To validate the augmented dataset, the quality of translation is evaluated using the Semantic Cosine similarity score and BERTScore.

2) Cosine similarity score: It is a semantic similarity metric. We use cosine similarity with LaBSE to verify the quality of the translated text. This verifies how much core meaning the translation retains. Because the proposed work uses Hindi, we use LaBSE to assess the validity of the back-translated text. It uses Eq. (19) to calculate cosine similarity.

$$\text{Cosine Similarity}(V_{orig}, V_{back}) = \frac{V_{orig} \cdot V_{back}}{\|V_{orig}\| \, \|V_{back}\|} \quad (19)$$

The average semantic similarity score using cosine similarity with LaBSE is 0.8450 for the implicitly balanced text and 0.7928 for the complete augmented dataset. The LaBSE cosine similarity scores of 0.8450 and 0.7928 indicate substantial semantic similarity between the original and back-translated text. High scores indicate that, despite phrasing variations, back-translation preserved the original text's meaning.

3) BERT score: BERT score is also a semantic similarity metric that computes sentence-level similarity. mBERT computes cosine similarity between sentences using contextualised token embeddings. It captures paraphrases and semantic equivalence. The final mBERT score is calculated using contextual embeddings for tokens in the original and back-translated sentences. The final F1 score measures semantic similarity between the original and back-translated sentence. A higher mBERT score indicates that the original text is semantically close to the final back-translation. The mBERT score for the back-translated text on the implicit balanced dataset is 0.8084, and on the complete augmented dataset it is 0.7824. This higher score indicates that the backtranslated text is semantically close to the original.

IV. Experiment

This section presents the experimental setup and the baseline and state-of-the-art (SOTA) methods used for comparison. Except for the BiGRU+Capsule Network, all other baseline and SOTA models were implemented and reproduced based on the architectural and experimental details reported in HHSD [3]. We followed the same HHSD dataset and preprocessing procedure for these reproduced baseline and SOTA experiments. The details of the same are as follows:

We compare the following baseline and state-of-the-art (SOTA) methods:

- LaBSE [25], [3]: It is used to produce vector embeddings for 109 languages using language-agnostic BERT.
- MuRIL[26], [3]: MuRIL was developed specifically for Indian languages. To improve performance on Indian languages, this model is trained on 16 of them.
- IndicBERT [27], [3]: It uses fewer parameters than some BERT variants. IndicBERT is a multilingual language model developed using data from 12 Indian languages.
- XLM-RoBERTa [28], [3]: This model was trained on 100 languages and demonstrates good cross-lingual classification performance compared to mBERT.
- mBERT [29], [3]: mBERT led the development of Multilingual BERT, a pre-trained transformer model supporting 104 languages.
- BiGRU + Capsule Network [30]: The BiGRU + Capsule Network [30] baseline consists of a 128-unit bidirectional GRU layer, followed by a capsule layer containing 16 capsules of dimension 32. The capsule layer uses five routing iterations. The capsule representations are aggregated using global average pooling, followed by a dropout layer (rate 0.3) and a two-class softmax output layer. We optimised the model using Adam with a learning rate of 0.001. Training was performed for 10 epochs with a batch size of 512.
- CNN [31]: The CNN baseline was implemented using a Conv1D layer with 100 filters and a kernel size of 4, followed by max pooling. A fully connected layer with 32 ReLU units followed before the output layer. The model was optimized with Adam at a learning rate of 0.001. Training was performed for 3 epochs with a batch size of 30.
- BiLSTM [32]: The input representation was processed by a BiLSTM with 100 hidden units, followed by MaxPooling1D with a pool size of 2. A fully connected layer with 32 ReLU units preceded the output layer. The model was optimized with Adam (learning rate 0.001), a batch size of 30, and 2 epochs of training.

A. Experimental Setup of Proposed Technique

We developed the proposed model in TensorFlow and initialized it using the FastText embedding technique. We used pre-trained 300-dimensional Hindi FastText vectors to initialize the embedding matrix. The embedding layer was kept trainable during model training to allow task-specific adaptation of the representations. Performance was evaluated with 5-fold cross-validation. Training used a fixed 5 epochs per fold; we optimized model with the Adam optimizer at a learning rate of 9e-5, keeping the remaining optimizer parameters at their default TensorFlow/Keras settings. We used binary cross-entropy as the objective function and trained with mini-batches of 8 samples. We selected the hyperparameter values through preliminary experiments on the training data, considering validation performance and convergence stability. We varied the learning rate, batch size, number of training epochs, and hidden-unit configurations over a small set of empirically reasonable values. We selected the configuration that provided stable validation performance. We did not perform exhaustive hyperparameter optimization. We used the same hyperparameters for both the fully augmented and balanced datasets. We used a fixed random seed of 42 to ensure reproducibility. Input sequences were padded or truncated to a maximum sentence length of 50 tokens. The proposed architecture employed 128 hidden units in the word-level BiLSTM and 64 hidden units in the sentence-level BiLSTM. We applied a dropout rate of 0.4 after the fully connected layer to reduce overfitting during training. The attention mechanism used trainable attention weights with softmax normalization to identify important contextual information from the input sequence. In the attention layer, the trainable attention weight matrix and attention score vector were initialized from a random-normal distribution, while the attention bias was initialised to zeros.

We conducted the experiments on an NVIDIA RTX A4000 GPU with 16 GB GPU memory and 32 GB system RAM. Across the 5-fold cross-validation, the average training time was 22.29±2.65 min per fold. The average model evaluation time across the five folds was 12.20±9.39 sec per fold.

B. Evaluation Metrics

We evaluate the proposed technique using accuracy and weighted F1. The accuracy is calculated using Eq. (20)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

Here, TP and TN stand for true positives and true negatives (successful hits and correct misses), while FP and FN denote false alarms and overlooked cases.

The dataset exhibits significant class imbalance; therefore, standard accuracy is not an appropriate evaluation metric, as it may produce inflated scores by favouring the majority class. For highly imbalanced datasets, the weighted F1-score is more appropriate because it averages F1 scores across all classes, assigning weights that reflect the number of true examples in each class. We evaluated model performance using the weighted F1-score metric Eq. (21).

$$\text{Weighted F1} = \frac{\sum_{i=1}^{N}(F1_{class\_i} \times S_{class\_i})}{\sum_{i=1}^{N} S_{class\_i}} \tag{21}$$

where, N: Total number of classes.

$F1_{class\_i}$: F1 score for class i.

S_{class_i}: Support (number of actual instances) for class i.

C. Results

On the original dataset, the model correctly identifies 301 samples from the implicit class (0) and 796 samples from the explicit class (1), as summarized in Fig. 3.

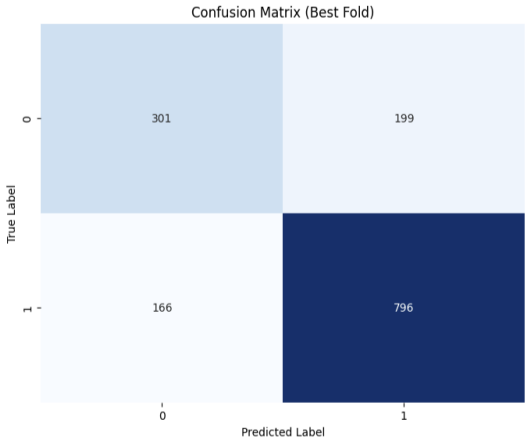


Fig. 3. Confusion matrix.

For the best-performing fold, the proposed model achieved an overall accuracy of 75.03% and a weighted F1-score of 74.81%. The per-class analysis shows that the model achieved 64.45% precision, 60.20% recall, and a 62.25% F1-score for implicit hate speech. In contrast, the corresponding values for explicit hate speech were 80.00%, 82.74%, and 81.35%, respectively. The macro F1-score was 71.80%. Table III presents the class-wise Precision, Recall, and F1-score for the best-performing fold.

These results reflect balanced predictive behaviour across both categories. The Best accuracy and loss of the model are shown in Fig. 4 and 5 for the original dataset. Test-set accuracy continued to improve through 5 epochs.

TABLE III. CLASS-WISE CLASSIFICATION PERFORMANCE FOR THE BEST-PERFORMING FOLD

Class	Precision (%)	Recall (%)	F1-score (%)	Support
Implicit Hate Speech (IHS)	64.45	60.20	62.25	500
Explicit Hate Speech (EHS)	80	82.74	81.35	962
Macro Average	**72.23**	**71.47**	**71.80**	1462
Weighted Average	**74.69**	**75.03**	**74.82**	1462

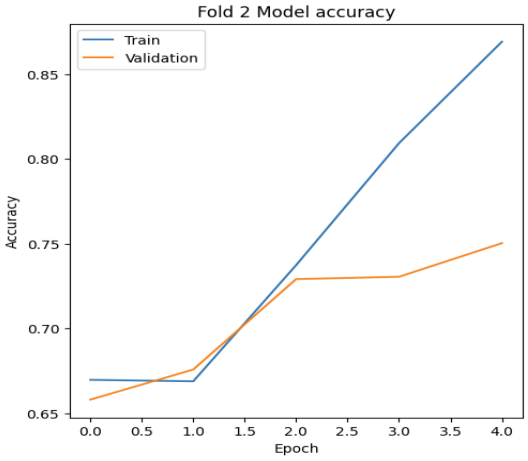


Fig. 4. Model accuracy.

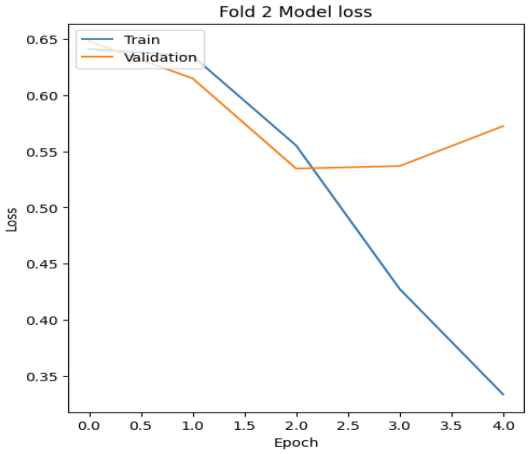


Fig. 5. Model loss.

Fig. 6 depicts the ROC curve obtained for the best validation fold. The curve's proximity to the upper-left region reflects the model's stronger ability to correctly distinguish explicit hate instances while keeping the false positive rate low. With an AUC (Area Under the Curve) of 0.71, the model shows a fair capacity to separate explicit from implicit hate speech. This value indicates a 71% chance that a randomly selected explicit hate instance will be ranked as explicit. Because detecting implicit hate is very difficult, an AUC of 0.71 is fair for classifying implicit/explicit HSD.

Table IV shows the number of parameters required for pretrained transformer models.

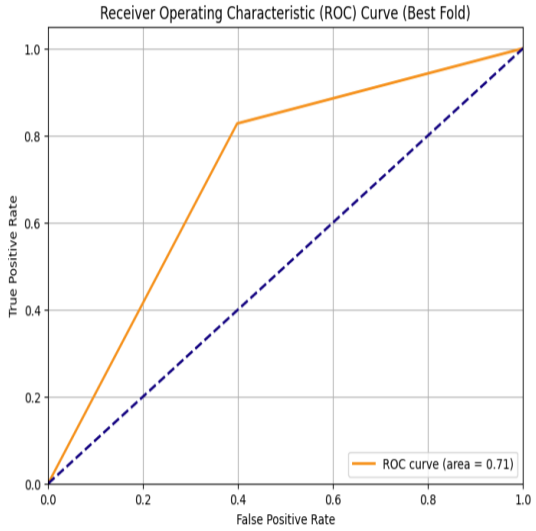


Fig. 6. ROC curve.

TABLE IV: Parameter comparison

Model	# Parameters
IndicBERT [27]	$\geq$ 18M
MuRIL [26]	236M
LaBSE [25]	$\geq$ 110M
XLM-RoBERTa [28]	270M
mBERT [29]	172M
Proposed model	**12M**

mBERT has 172 million parameters, while XLM-RoBERTa has approximately 270 million, which is very large. This large parameter count implies a larger model footprint and potentially greater computational requirements. The proposed model requires approximately 12 million parameters, resulting in a substantially smaller parameter footprint. Thus, the proposed architecture is relatively lightweight in terms of parameter count.

This method outperforms CNN, BiLSTM, BiGRU+Capsule Network, IndicBERT, MuRIL, LaBSE, XLM-RoBERTa, and mBERT. The proposed model achieves a best accuracy of 75.03%, which is 2.51% higher than mBERT. The best weighted F1-score is 74.81%, which is 17.84% higher than that of mBERT.

Table V reports the accuracy and weighted F1-scores

achieved by the proposed method.

TABLE V: Result on implicit HHSD

Model	Accuracy	Weighted-F1
CNN [31]	66.10%	52.20%
BiLSTM [32]	68.72%	54.30%
BiGRU + Capsule Network [30]	69.92%	53%
IndicBERT [27]	66.50%	52.80%
MuRIL [26]	70.95%	56.33%
LaBSE [25]	69.17%	54.54%
XLM-RoBERTa [28]	66.30%	40.50%
mBERT [3]	72.52%	56.97%
Proposed	**75.03%**	**74.81%**

D. Discussion of Key Findings

1) Finding 1: The proposed model achieves the highest weighted F1-score (74.81%), outperforming mBERT by 17.84%.

2) Finding 2: The small gap between Accuracy (75.03%) and Weighted F1-score (74.81%) indicates that the overall performance is reasonably consistent when accounting for class-wise F1 contributions. However, the lower F1-score for implicit hate speech demonstrates that this class remains more challenging to classify.

3) Finding: The proposed hierarchical contextual framework achieves competitive performance against the evaluated transformer-based baselines, while class-wise results indicate that implicit hate remains harder to distinguish than explicit hate.

4) Finding 4: Semantic-preserving augmentation improves performance, while class balancing provides an additional improvement in weighted F1-score.

E. Ablation Study

The proposed work investigates the impact of back-translation-based augmentation and class balancing specifically for implicit hate speech detection, where contextual ambiguity and limited annotated data significantly affect classification performance. We ran experiments using 5-fold cross-validation and computed the mean weighted F1-score and standard deviation across folds. Results in Table VI indicate that back-translation-based augmentation improves contextual diversity and enhances model generalization. Furthermore, balancing implicit and explicit hate samples significantly improves classification stability and minority class detection performance. The balanced augmented dataset achieved the best overall performance with the highest mean weighted F1-score.

Augmenting the complete dataset further enhances model performance. Experimental results show that data augmentation improves model performance compared with training on the original dataset.

TABLE VI. Ablation Study Using 5-Fold Cross-Validation

Variant	Weighted F1-score (Mean $\pm$ Std)
Proposed	73.33 ± 0.97
Proposed+augmented	74.95 ± 0.55
Proposed+augmented balanced	79.04 ± 1.44

In 5-fold cross-validation, the best fold achieves a weighted F1-score of 75.81% and an accuracy of 76.42%. These results represent an 18.84% improvement in weighted F1-score and a 3.9% increase in accuracy compared to mBERT. Relative to the original dataset, the weighted F1-score increases by 1% and accuracy by 1.39%.

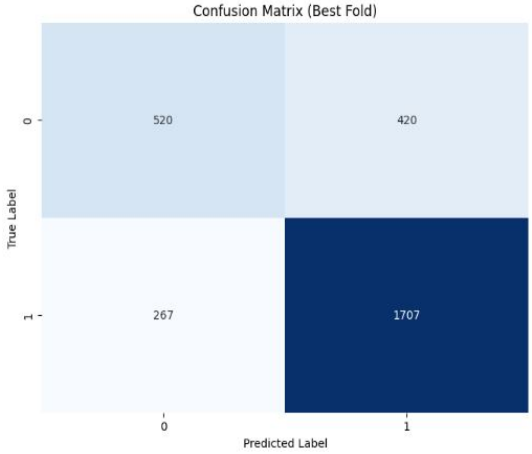


Fig. 7. Confusion matrix of complete augmentation.

Fig. 7 presents the confusion matrix, Fig. 8 displays the accuracy and Fig. 9 illustrates the ROC curve for the fully augmented dataset.

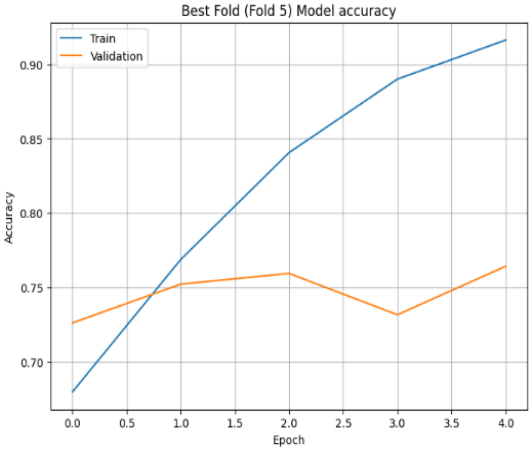


Fig. 8. Accuracy of complete augmentation.

To correct the dataset imbalance, we used data augmentation with backtranslation, as described in Section III-K. Only implicit data was augmented for balancing. The model was then evaluated on the balanced dataset, which showed significantly improved performance compared to the imbalanced

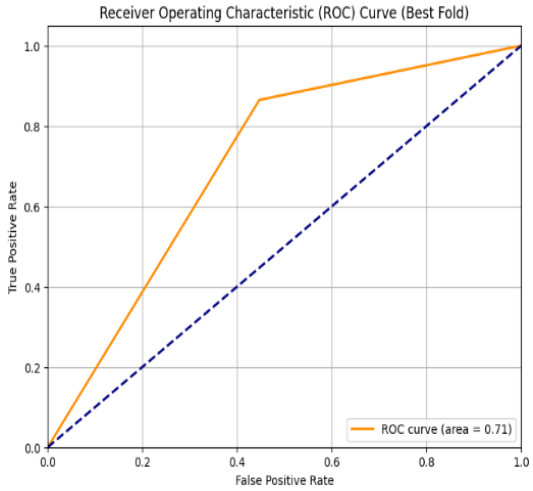


Fig. 9. ROC curve of complete augmentation.

version. In 5-fold cross-validation, the highest weighted F1-score of 80.70% and accuracy of 80.70% were achieved, reflecting improvements of 8.18% and 23.73% over mBERT. The weighted F1-score increased by 5.89% and accuracy by 5.67% compared to the original dataset. The balanced dataset is evaluated using multiple performance indicators. Class-wise results, accuracy trends, and ROC analysis are shown in Fig. 10, 11 and 12.

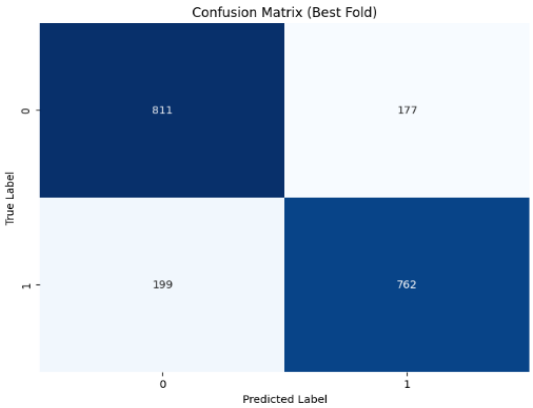


Fig. 10. Confusion matrix of balanced dataset.

F. Error Analysis

The study provides detailed qualitative error analysis demonstrating how contextually sensitive lexical terms can confuse deep learning models between implicit and explicit hate speech categories. The proposed model's confusion matrix shows that many implicit hate speech instances are misclassified. Unlike explicit hate speech, implicit hate often lacks direct offensive vocabulary or explicit slurs and instead conveys hateful intent through indirect references, sarcasm, stereotypes, or hidden contextual meaning. These linguistic characteristics make implicit hate speech much harder to identify. In addition, the relatively smaller number of implicit hate samples in the dataset also affects the model's classification capability and contributes to prediction bias toward the majority class.

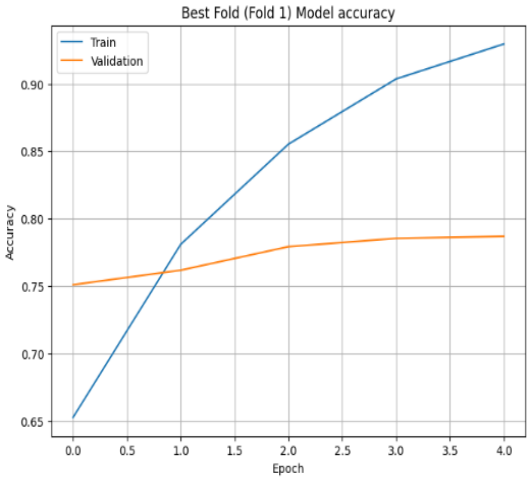


Fig. 11. Accuracy of balanced dataset.

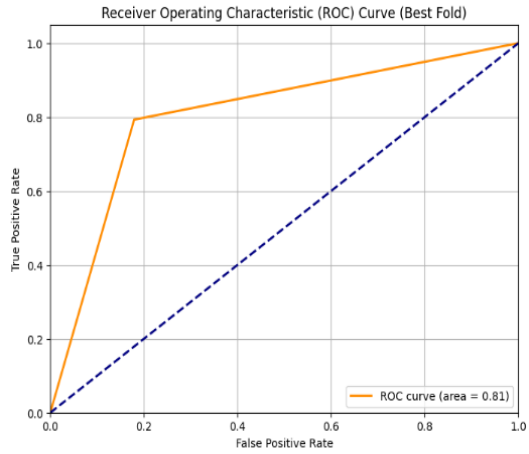


Fig. 12. ROC curve of balanced augmentation.

For example, the statement "माफ करिए 🙏 पैसे देकर खरीदते हैं हराम का तो नही खाते" (Translation: *"Sorry 🙏, we buy it with our own money; we do not consume anything forbidden or unlawful."*) was misclassified as explicit hate speech due to the presence of contextually sensitive terms such as हराम(forbidden). Although the statement does not contain direct abusive language, the implicit sarcastic tone and indirect targeting may cause the model to associate such terms with explicit hateful expressions. This demonstrates the difficulty of distinguishing implicit hate from explicit hate when contextual and cultural understanding is required.

The proposed model correctly identified most explicit hate speech instances; however, it still misclassified a small number of samples. Explicit hate speech detection primarily relies on the presence of offensive vocabulary, abusive expressions, and explicit slurs, which are generally easier for the model to recognize. However, in some cases, abusive or sensitive words are used indirectly within an implicit contextual setting rather than as direct hate expressions. In such situations, the model tends to associate these terms with explicit hate speech and may incorrectly classify implicit hate instances as explicit hate.

For instance, the statement "इतिहास में आज जब अंटार्कटिका पर पहली बार पड़े इंसान के कदम साथ में कुत्ते भी वहां पहुंचे थे" (Translation: *"On this day in history, when humans first stepped onto Antarctica, dogs had also reached there along with them."*)was an implicit hate speech instance. Still, the model incorrectly classified it as explicit hate speech. The misclassification may have occurred because of the word "कुत्ते" (dog), which is frequently associated with abusive or offensive expressions in hateful content. Although the statement uses the term indirectly rather than as a direct insult, the model likely relied on lexical associations and interpreted it as explicit hate speech. This shows that contextually sensitive words can confuse the model when distinguishing between implicit and explicit hate speech.

V. Conclusion and Future Work

The proliferation of hateful discourse on social media platforms poses increasing risks to social harmony. It highlights the need for effective automated Hate Speech Detection (HSD), particularly for low-resource languages such as Hindi. This study developed a hierarchical BiLSTM–attention-based framework for Hindi hate speech detection, emphasizing the identification of both implicit and explicit forms of hate.

The principal contributions of this study include the task-specific integration of a hierarchical BiLSTM architecture with attention-based feature selection for contextual discrimination of implicit and explicit hate speech, the selective retention of stop words and emojis to preserve contextual and emotional information, and the incorporation of semantic-preserving data augmentation and class balancing to address class imbalance and increase linguistic diversity. The proposed framework does not rely on large transformer-based language models and has an approximately 12 million parameter footprint, compared with approximately 172 million parameters for mBERT.

Its ability to identify both implicit and explicit hate expressions can help human moderators prioritize potentially harmful content for further review. Thus, the proposed approach provides a parameter-efficient alternative for Hindi hate speech detection while retaining the ability to model hierarchical contextual information.

The present study has some limitations. First, the experimental evaluation is conducted on the HHSD dataset. The dataset's characteristics, including its topic distribution, annotation scheme, and domain-specific expressions, may influence model performance and limit the generalizability of the findings. Second, publicly available Hindi datasets generally use different annotation schemes, including hate/non-hate, hate/offensive/profane, hostility-related, or target-based categories. Because equivalent implicit/explicit hate-speech annotations are limited, directly evaluating this specific task on other Hindi corpora is challenging. Third, the applicability of the proposed approach to multilingual settings and cross-domain environments requires further investigation.

Future research will evaluate the proposed approach on additional Hindi corpora, cross-domain datasets, and multilingual resources to assess its robustness and generalizability. Future research will also explore integrating contextual and region-specific cultural knowledge to improve model performance further.

Overall, the proposed work shows that hierarchical contextual representation learning can effectively support Hindi hate speech detection, including identifying implicit hate expressions, without relying solely on large transformer-based architectures. The combination of hierarchical context modelling, semantic-preserving augmentation, class balancing, and parameter efficiency provides a practical foundation for developing more robust and deployable Hindi hate speech detection systems.

Declarations

Ethics Approval and Consent to Participate

This study does not involve human participants or animal subjects requiring ethical approval.

Competing Interests

The authors declare no conflict of interest.

References

[1] United Nations, "What is hate speech? understanding hate speech," https://www.un.org/en/node/165070, 2025, accessed: July 4, 2025.

[2] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing," *Neurocomputing*, vol. 546, p. 126232, 2023.

[3] P. Kapil, G. Kumari, A. Ekbal, S. Pal, A. Chatterjee, and B. N. Vinutha, "HHSD : Hindi hate speech detection leveraging multi-task learning," *IEEE Access*, vol. 11, pp. 101 460–101 473, 2023.

[4] M. ElSherief, C. Ziems, D. Muchlinski, V. Anupindi, J. Seybolt, M. De Choudhury, and D. Yang, "Latent hatred: A benchmark for understanding implicit hate speech," Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, pp. 345–363, 2021. [Online]. Available: https://aclanthology.org/2021.emnlp-main.29/

[5] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[6] J. A. Siddiqui, S. S. Yuhaniz, and Z. A. Memon, "Unmasking online hostility: Analysing and mitigating hate speech in social media," *Baghdad Science Journal*, vol. 22, no. 4, pp. 1393–1408, 2025.

[7] A. M. U. D. Khanday, Q. R. Khan, and S. T. Rabani, "Detecting textual propaganda using machine learning techniques," *Baghdad Science Journal*, vol. 18, no. 1, pp. 199–209, 2021.

[8] A. Sharma, A. Kabra, and M. Jain, "Ceasing hate with moh: Hate speech detection in hindi–english code-switched language," *Information Processing & Management*, vol. 59, no. 1, p. 102760, 2022.

[9] A. Z. Miran and H. S. Yahia, "Hate speech detection in social media (twitter) using neural network," *Journal of Mobile Multimedia*, vol. 19, no. 3, pp. 765–798, 2023.

[10] M. Subramanian, V. Easwaramoorthy, G. Deepalakshmi, and J. Cho, "A survey on hate speech detection and sentiment analysis using machine learning and deep learning models," *Alexandria Engineering Journal*, vol. 80, pp. 110–121, 2023.

[11] A. Kamal, T. Anwar, V. K. Sejwal, and M. Fazil, "Bicapshate: Attention to the linguistic context of hate via bidirectional capsules and hatebase," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1781–1792, 2024.

[12] R. Panchal, P. Shelke, S. Ravandale, and S. Patil, "Shielding social media: Detection of toxic memes for automated moderation," International Conference on Sustainability, Innovation & Technology (ICSIT), pp. 1–6, 2025, 10.1109/ICSIT65336.2025.11295293.

[13] P. Kapil and A. Ekbal, "Transformer-based ensemble learning for hate speech detection leveraging sentiment and emotion knowledge sharing," Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, pp. 4485–4496, 2022.

[14] Y. Shukla, R. V. Panchal, T. Nigade, S. Khodade, and P. Pimpalkar, "A multilingual bert-based framework for robust online hate speech detection," International Conference on Computer Vision and Machine Intelligence (CVMI), pp. 1–8, 2024, 10.1109/CVMI61877.2024.10782783.

[15] A. Velankar, H. Patil, A. Gore, S. Salunke, and R. Joshi, "L3cube-mahahate: A tweet-based marathi hate speech detection dataset and bert models," 2022. [Online]. Available: https://arxiv.org/abs/2203.13778

[16] V. Ibrahim, J. A. Bakar, N. H. Harun, and A. F. Abdulateef, "A word cloud model based on hate speech in an online social media environment," *Baghdad Science Journal*, vol. 18, no. 2, pp. 937–944, 2021.

[17] Y. Kim, S. Park, and Y.-S. Han, "Generalizable implicit hate speech detection using contrastive learning," Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, pp. 6667–6679, 2022. [Online]. Available: https://aclanthology.org/2022.coling-1.579/

[18] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Hate speech detection and racial bias mitigation in social media based on bert model," *PLOS ONE*, vol. 15, no. 8, p. e0237861, 2020.

[19] F. Barbieri, F. Ronzano, and H. Saggion, "What does this emoji mean? a vector space skip-gram model for twitter emojis," Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, pp. 3967–3972, 2016. [Online]. Available: https://aclanthology.org/L16-1626/

[20] W. Ashiq, S. Kanwal, A. Rafique, M. Waqas, T. Khurshaid, E. Caro Montero, A. Bustamante Alonso, and I. Ashraf, "Roman urdu hate speech detection using hybrid machine learning models and hyperparameter optimization," *Scientific Reports*, vol. 14, p. 28590, 2024.

[21] Q. Sifak and E. B. Setiawan, "Hate speech detection using cnn and bigru with attention mechanism on twitter," International Conference on Communication, Networks and Satellite (COMNETSAT), pp. 170–175, 2023, 10.1109/COMNETSAT59769.2023.10420628.

[22] A. R. Jafari, G. Li, P. Rajapaksha, R. Farahbakhsh, and N. Crespi, "Fine-grained emotions influence on implicit hate speech detection," *IEEE Access*, vol. 11, pp. 105 330–105 343, 2023.

[23] K. Maity, A. S. Poornash, S. Bhattacharya, S. Phosit, S. Kongsamlit, S. Saha, and K. Pasupa, "Hatethaisent: Sentiment-aided hate speech detection in thai language," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 5, pp. 5714–5727, 2024.

[24] K. Chavinda, P. Kalansooriya, T. Weerasuriya *et al.*, "An emotion aware context adaptive machine learning approach for detecting hate speech in social media," *Neural Computing and Applications*, vol. 38, p. 268, 2026.

[25] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic bert sentence embedding," Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, pp. 878–891, 2022. [Online]. Available: https://aclanthology.org/2022.acl-long.62/

[26] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar, "Muril: Multilingual representations for indian languages," 2021. [Online]. Available: https://arxiv.org/abs/2103.10730

[27] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, and P. Kumar, "Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages," Findings of the Association for Computational Linguistics: EMNLP 2020, Online, pp. 4948–4961, 2020. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.445/

[28] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp. 8440–8451, 2020. [Online]. Available: https://aclanthology.org/2020.acl-main.747/

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186, 2019. [Online]. Available: https://aclanthology.org/N19-1423/

[30] P. Rathnayaka, S. Abeysinghe, C. Samarajeewa, I. Manchanayake, and M. Walpola, "Sentylic at iest 2018: Gated recurrent neural network and capsule network based approach for implicit emotion detection," Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, pp. 254–259, 2018. [Online]. Available: https://aclanthology.org/W18-6237/

[31] Y. Kim, "Convolutional neural networks for sentence classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1746–1751, 2014. [Online]. Available: https://aclanthology.org/D14-1181/

[32] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, 2005.

TrustEdge-AI: Uncertainty-Aware Dynamic Trust Intelligence for Adaptive Cyber Defense in IoT Edge Networks

Mohamed Ayari[1]*, Ahmed Alhomoud[2]*, Atef Gharbi[3], Yamen El Touati[4], Zeineb Klai[5]
Department of Information Technology-Faculty of Computing and Information Technology,
Northern Border University, Saudi Arabia[1]
Department of Computer Science, College of Science, Northern Border University, Arar 91431, Saudi Arabia[2]
Department of Information Systems-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91911, Saudi Arabia[3]
Department of Computer Sciences-Faculty of Computing and Information Technology,
Northern Border University, Saudi Arabia[4,5]

***Abstract*—Conventional Internet-of-Things (IoT) intrusion detection typically maps a model score directly to an alert, although autonomous defense must also determine whether the evidence is reliable, whether suspicious behavior persists over time, and whether a disruptive response is justified. This study proposes TrustEdge-AI, an uncertainty-aware dynamic-trust framework for adaptive cyber defense at the IoT edge. For each monitored device, the framework converts anomaly and classification outputs into reliability-weighted evidence, fuses this evidence with temporal behavioral consistency and optional neighborhood context, and maintains a bounded trust state that evolves over successive monitoring intervals. A hysteretic policy then maps trust regions to graduated actions: allow, observe, restrict, and isolate/escalate. The formulation explicitly separates predictive confidence, uncertainty, instantaneous risk, accumulated trust, and enforcement authority. Analytical trust-state trajectories derived from the proposed update equations demonstrate the intended response to stable benign behavior, uncertain drift, sustained compromise, and post-incident recovery. In addition, a leakage-safe evaluation protocol is specified for BoT-IoT, TON-IoT, and UNSW-NB15, covering known-attack detection, withheld-family stress testing, calibration, trust dynamics, ablation, latency, and communication overhead. Experimental evaluation on BoT-IoT, TON-IoT, and UNSW-NB15 demonstrates strong detection performance, improved calibration, low false-restriction rates, stable policy behavior, and low-latency edge operation. The results further show that uncertainty weighting, temporal trust, contextual fusion, and hysteresis contribute complementary gains to decision reliability and response stability.**

***Keywords*—Internet of things; edge intelligence; intrusion detection; dynamic trust; uncertainty estimation; adaptive cybersecurity; anomaly detection; calibration**

I. Introduction

The Internet of Things (IoT) increasingly connects sensors, cameras, meters, controllers, healthcare devices, and industrial assets to edge and cloud services. This connectivity enables automation but expands the attack surface of heterogeneous and resource-constrained endpoints. IoT deployments face scanning, credential abuse, botnet recruitment, denial-of-service, malware, data injection, and lateral movement, while centralized analysis can introduce communication cost and response delay [1]–[3]. Edge computing addresses part of this problem by moving selected analytics closer to the data source [4], [5]. However, low-latency inference does not by itself make autonomous mitigation trustworthy.

Most machine-learning intrusion detection systems answer a classification question: is the current observation benign or malicious? Operational defense requires a richer decision. A high anomaly score can represent a real attack, a legitimate workload change, sensor noise, or distribution shift. Likewise, a compromised endpoint can remain only moderately anomalous while its behavior changes gradually. A response engine that treats every model output as equally authoritative can therefore overreact to uncertain alarms or underreact to persistent low-intensity behavior.

A useful defense mechanism should preserve memory of previous evidence and explicitly represent the reliability of current predictions. Trust management provides such a state variable, while uncertainty estimation provides a mechanism for reducing the authority of unreliable predictions. Prior work has explored edge IDS [6], IoT botnet detection [7], collaborative trust management [8], trust-centric edge security [9], IoT–fog trust [10], holistic trust-based architectures [11], and federated trust-based access control [12]. Nevertheless, calibrated predictive uncertainty is rarely coupled directly to a temporal device-trust state that governs graduated mitigation.

This study proposes *TrustEdge-AI*. Each monitored device d_i maintains a trust state $T_i(t) \in [0,1]$. Edge inference produces anomaly evidence and class probabilities; predictive uncertainty discounts unreliable evidence; temporal and neighborhood terms capture persistence and coordinated behavior; and a policy layer converts trust regions into graduated actions. The framework therefore changes the semantics of autonomous defense from *prediction* $\rightarrow$ *action* to *prediction* $\rightarrow$ *reliable evidence* $\rightarrow$ *risk* $\rightarrow$ *temporal trust* $\rightarrow$ *action*.

*Corresponding author.

The contributions are:

- an uncertainty-aware evidence model that distinguishes classifier confidence from the reliability of using that confidence for enforcement;
- a bounded dynamic-trust update combining current evidence, temporal persistence, optional neighborhood context, controlled recovery, and hysteresis;
- an interpretable trust-region policy supporting graduated response and device-criticality constraints;
- analytical trust-state analysis that exposes the effect of sustained compromise, uncertainty, memory, and recovery; and
- a leakage-safe, multi-dataset evaluation protocol for BoT-IoT, TON-IoT, and UNSW-NB15 with calibration, withheld-family stress testing, ablation, latency, and overhead reporting.

II. Related Work and Research Gap

A. IoT Intrusion Detection at the Edge

BoT-IoT provides botnet-oriented IoT traffic for network-forensic analytics [13]; TON-IoT broadens evaluation through heterogeneous IoT/IIoT telemetry and network data [14]; and UNSW-NB15 remains a complementary benchmark containing multiple contemporary network-attack categories [15]. These datasets have supported supervised, anomaly-based, and deep IDS research. Passban IDS demonstrated anomaly-based detection for IoT edge devices [6], while N-BaIoT showed the value of deep autoencoders for IoT botnet detection [7]. Such systems primarily optimize detection quality, whereas TrustEdge-AI focuses on the decision layer that determines how repeated and uncertain evidence should affect device status.

B. Trust Management

Trust mechanisms estimate whether devices, peers, communications, or services should be relied upon. Collaborative IDS with trust management has been proposed for distributed IoT [8]; trust-centric detection has been considered for medical IoT edge networks [9]; and two-way IoT–fog trust has been studied for task offloading [10]. SHIELD combines monitoring, IDS, ledger mechanisms, and trust management [11]. Recent work also considers instantaneous communication trust [16] and federated edge intelligence with trust-based access control [12]. These contributions establish trust as an important security abstraction, but trust is often associated with reputation, access control, communication quality, or collaboration rather than being updated from uncertainty-weighted intrusion evidence.

C. Predictive Uncertainty and Calibration

A model probability is not automatically a calibrated probability. Temperature scaling can improve correspondence between confidence and empirical correctness [17], while deep ensembles offer a practical approach to predictive uncertainty estimation [18]. Recent work involving the present authors has also demonstrated cross-domain transfer learning with explainability for malware and IoT/network intrusion detection on NF-TON-IoT and UNSW-NB15 [19], as well as transformer-based anomaly detection for resource-constrained IoT devices [20]. These studies strengthen the case for capable edge-security analytics, whereas TrustEdge-AI addresses the complementary decision-control problem of determining how uncertain and temporally persistent evidence should govern mitigation. In autonomous defense, uncertainty therefore has a direct operational meaning: an uncertain prediction should exert less influence on a disruptive action than repeated, consistent, high-confidence evidence.

D. Positioning

TrustEdge-AI is not proposed as another classifier; it is a stateful decision architecture placed between detection and enforcement. The comparison with prior work is based on four explicit capabilities: 1) execution of security analytics at or near the edge, 2) a device state that evolves from successive observations, 3) explicit attenuation of evidence according to predictive uncertainty, and 4) more than a binary alert/no-alert response. Existing edge IDS primarily address the first capability [6], [7]; trust-management studies provide temporal or reputation state but do not generally couple that state to calibrated intrusion uncertainty [8]–[10]; and federated trust/access-control work combines edge intelligence and trust but targets access control rather than uncertainty-weighted graduated intrusion response [12]. TrustEdge-AI is designed specifically to combine these four capabilities in a single decision layer. This textual positioning replaces qualitative full/partial/absent scores whose interpretation can otherwise be subjective.

III. System and Threat Model

Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ denote IoT devices connected through gateways to one or more edge nodes. During monitoring interval t, device d_i generates feature vector $\mathbf{x}_i(t) \in \mathbb{R}^p$. Preprocessing produces $\mathbf{z}_i(t) = \phi(\mathbf{x}_i(t))$. An edge detector returns an anomaly score $A_i(t) \in [0, 1]$ and class posterior vector $\mathbf{p}_i(t)$.

The adversary may compromise endpoints, generate DDoS traffic, scan services, abuse credentials, inject malicious traffic, or move laterally. The base model assumes that the edge execution environment and model repository are trusted. Model poisoning, edge-node physical compromise, and model extraction are outside the claimed protection boundary. This assumption is made explicit because neighborhood evidence and policy decisions become unsafe if the edge trust anchor itself is malicious.

For every device, TrustEdge-AI maintains:

$$T_i(t) \in [0, 1], \tag{1}$$

where, one denotes strong current trust and zero denotes a highly untrusted state. $T_i(t)$ is not an identity credential, a calibrated probability of compromise, or an expected-loss estimate. It is a bounded operational security state that summarizes the recent history of reliability-weighted risk. Its semantics are therefore ordinal and policy-oriented: lower values indicate

stronger accumulated evidence for restrictive action, while the thresholds in Section IV-E map ranges of this state to auditable response levels. This distinction prevents probabilistic meaning from being assigned to a quantity that is defined by a recursive control law rather than by probabilistic calibration.

IV. Proposed TrustEdge-AI Framework

A. Architecture and Separation of Concerns

Fig. 1 separates the analytics/evidence plane from the trust/policy plane. This separation is deliberate. The edge IDS can be replaced without changing the trust semantics, while policy can be adapted to device criticality without retraining the detector.

The architecture has four logical functions: 1) edge detection, 2) uncertainty-aware evidence formation, 3) temporal trust/risk fusion, and 4) adaptive policy. The key design choice is the boundary between evidence and authority: a detector can state that an observation appears malicious, but the trust layer determines how much that statement should alter the persistent device state.

B. Uncertainty-Aware Intrusion Evidence

Let malicious-class confidence be:

$$C_i(t) = \max_{k \in \mathcal{K}_{\text{mal}}} p_{ik}(t). \tag{2}$$

In the reported implementation, predictive uncertainty $U_i(t) \in [0,1]$ is computed from the normalized entropy of the calibrated class-posterior vector. This choice is deterministic, inexpensive at the edge, and avoids ambiguity about the uncertainty estimator used in the experiments. For K classes,

$$U_i(t) = -\frac{1}{\log K} \sum_{k=1}^{K} p_{ik}(t) \log p_{ik}(t). \tag{3}$$

The raw evidence is:

$$Q_i(t) = \alpha A_i(t) + (1-\alpha) C_i(t), \qquad 0 \le \alpha \le 1, \tag{4}$$

and reliable evidence is:

$$E_i(t) = [1 - U_i(t)] Q_i(t). \tag{5}$$

Eq. (5) has an interpretable safety property: for fixed raw evidence Q_i, $\partial E_i / \partial U_i = -Q_i \le 0$. Thus increasing uncertainty can never increase the authority of the current malicious prediction.

C. Temporal and Neighborhood Context

Persistent deviation is more informative than an isolated spike. We maintain an exponentially weighted behavioral state

$$B_i(t) = \eta B_i(t-1) + (1-\eta) A_i(t), \qquad 0 \le \eta < 1. \tag{6}$$

When a meaningful device neighborhood $\mathcal{N}_i$ exists, contextual evidence is

$$N_i(t) = \frac{\sum_{j \in \mathcal{N}_i} w_{ij} E_j(t)}{\sum_{j \in \mathcal{N}_i} w_{ij} + \epsilon}. \tag{7}$$

The neighborhood term is optional rather than mandatory. This avoids creating artificial dependencies when the deployment has no defensible relationship graph. The behavioral memory $B_i(t)$ and trust recursion $T_i(t)$ serve different roles: $B_i(t)$ smooths only the anomaly channel before risk fusion, whereas $T_i(t)$ integrates the complete fused risk after uncertainty weighting and optional neighborhood context. Consequently, the two recursions operate at different stages and are not algebraically interchangeable temporal filters.

D. Instantaneous Risk and Dynamic Trust

Instantaneous risk is

$$R_i(t) = \beta_1 E_i(t) + \beta_2 B_i(t) + \beta_3 N_i(t), \quad \beta_m \ge 0, \quad \sum_m \beta_m = 1. \tag{8}$$

The default trust recursion is

$$T_i(t+1) = \rho T_i(t) + (1-\rho)[1 - R_i(t)], \quad 0 \le \rho < 1. \tag{9}$$

Because $R_i(t) \in [0,1]$ and $T_i(0) \in [0,1]$, Eq. (9) preserves $T_i(t) \in [0,1]$. For constant risk $R_i(t) = r$, the recursion converges to

$$\lim_{t \to \infty} T_i(t) = 1 - r, \tag{10}$$

while ρ controls the convergence rate. Hence ρ has a direct operational interpretation: high ρ makes trust resistant to short-lived evidence but slower to react to sustained compromise.

Controlled recovery is enabled only when $R_i(t) < \tau_r$ for L consecutive intervals:

$$T_i(t+1) = \min\{1, T_i(t) + \kappa[1 - T_i(t)]\}. \tag{11}$$

This condition prevents a single benign-looking interval from immediately restoring a previously compromised device. Once the recovery gate remains active, Eq. (11) has the closed-form relation

$$1 - T_i(t+n) = (1-\kappa)^n [1 - T_i(t)], \tag{12}$$

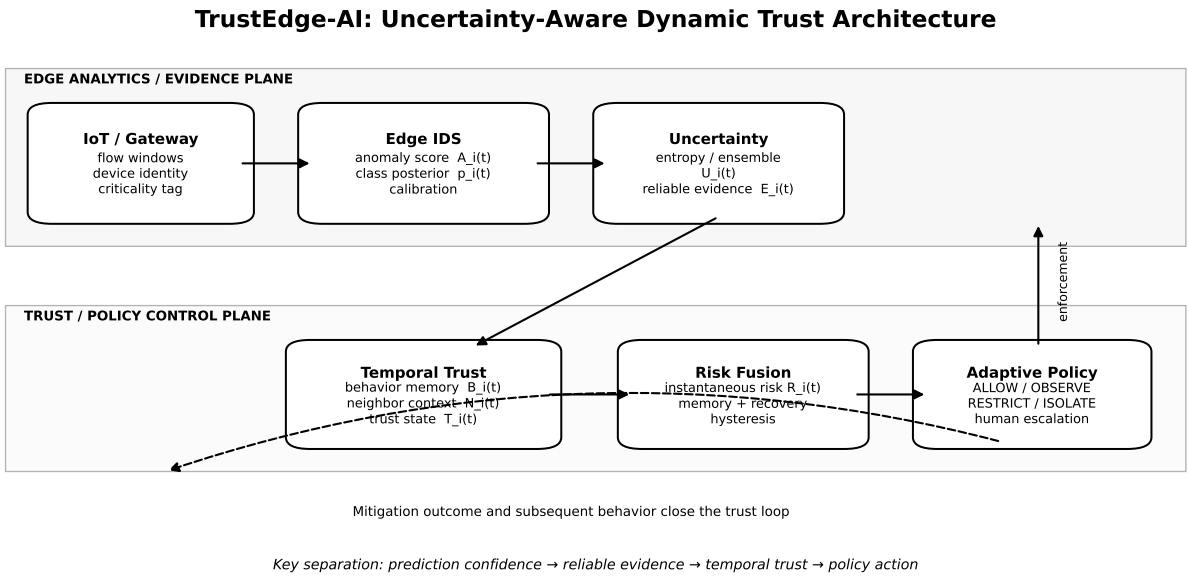


Fig. 1. TrustEdge-AI architecture. Detection outputs are not sent directly to enforcement. They are calibrated, converted to reliable evidence, accumulated into a dynamic trust state, and only then mapped to a policy action. The feedback path makes post-mitigation behavior part of subsequent trust evolution.

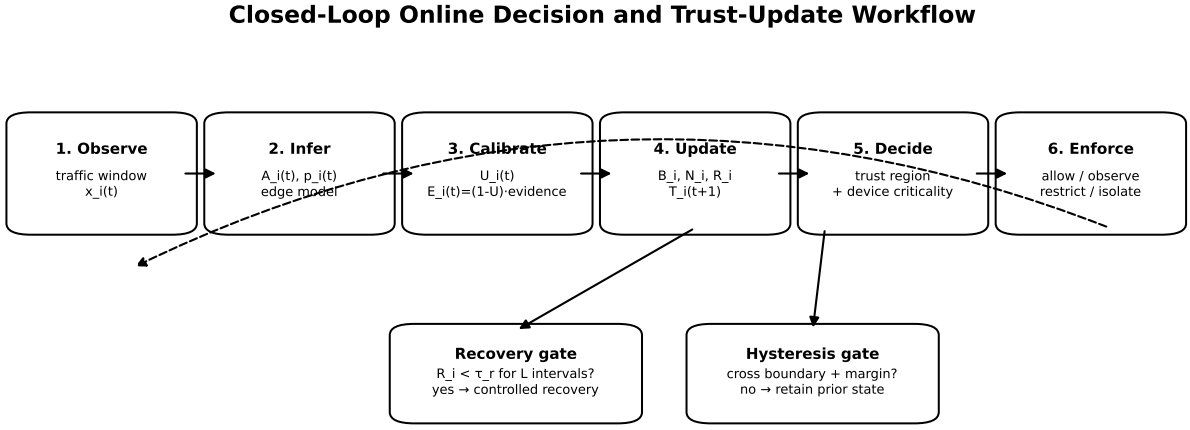


Fig. 2. Closed-loop online workflow. Recovery and hysteresis are explicit decision gates rather than implicit implementation details. This makes the conditions for trust restoration and policy-state transitions auditable.

so for $0 < \kappa \leq 1$ the recovery phase converges monotonically to $T_i = 1$. For a target recovery level $T^* < 1$, the number of active recovery updates satisfies

$$n \geq \frac{\log[(1-T^*)/(1-T_i(t))]}{\log(1-\kappa)}, \tag{13}$$

with the usual limiting interpretation at $\kappa = 1$. Hysteresis then determines whether this recovered trust is sufficient to cross a policy boundary. Thus recovery convergence and policy switching are analytically distinct.

E. Trust Regions, Hysteresis, and Criticality

Let $\tau_A > \tau_O > \tau_I$. The nominal policy is

$$\Pi(T_i) = \begin{cases} \text{Allow}, & T_i \geq \tau_A, \\ \text{Observe}, & \tau_O \leq T_i < \tau_A, \\ \text{Restrict}, & \tau_I \leq T_i < \tau_O, \\ \text{Isolate/Escalate}, & T_i < \tau_I. \end{cases} \tag{14}$$

A hysteresis margin $h > 0$ is applied around each boundary: entry into a more severe state requires crossing the lower boundary, whereas return requires crossing the corresponding boundary plus h. This reduces policy oscillation caused by small trust fluctuations. Device criticality is applied as a policy constraint; for example, a safety-critical controller can require human confirmation before isolation even when its trust state enters the most severe region.

Fig. 2 clarifies two aspects missing from a conventional

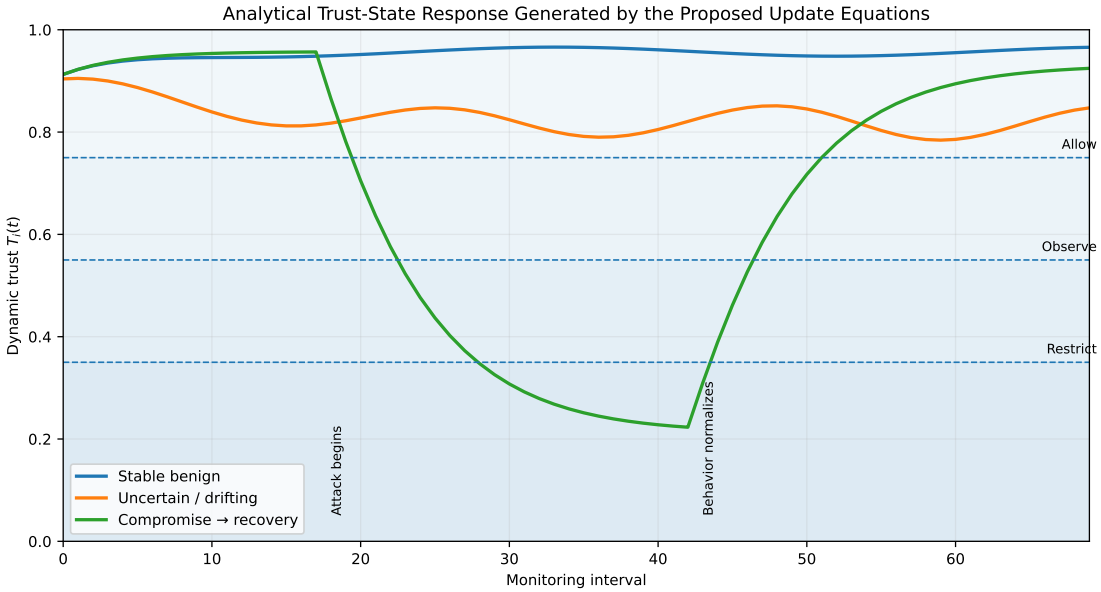


Fig. 3. Analytical trust-state response generated by the proposed equations. The stable benign sequence remains in the high-trust region; uncertain/drifting evidence causes limited degradation because uncertainty discounts current evidence; sustained compromise produces progressive trust loss; and normalized behavior permits gradual recovery. The curves are equation-driven illustrations, not benchmark measurements.

feed-forward IDS. First, mitigation does not terminate inference: subsequent behavior is fed back to the trust engine. Second, recovery and hysteresis are separate gates. The former determines whether trust may increase after an incident; the latter determines whether an increase is sufficient to change the operational policy state.

F. Online Procedure

For each device and monitoring interval, the edge node executes the following sequence: preprocess the observation; compute $A_i(t)$ and $\mathbf{p}_i(t)$; calibrate/estimate $U_i(t)$; form $E_i(t)$; update $B_i(t)$ and optional $N_i(t)$; compute $R_i(t)$; update $T_i(t+1)$; apply hysteresis and criticality constraints; enforce/log the selected action; and use subsequent observations to determine further degradation or controlled recovery. The computational overhead of the trust layer itself is $O(|\mathcal{N}_i|)$ per device when neighborhood aggregation is enabled and $O(1)$ otherwise, excluding the base detector.

V. Analytical Behavior and Interpretation

A methodological paper should demonstrate that its state equations behave as claimed before benchmark performance is discussed. Fig. 3 therefore plots analytical trajectories generated directly from Eq. (5)–(9) under three controlled evidence sequences. These curves are not empirical IDS results.

The stable benign trajectory demonstrates resistance to small fluctuations. The uncertain/drifting trajectory illustrates the role of Eq. (5): high uncertainty reduces the contribution of a single ambiguous prediction, so the system tends toward observation rather than immediate isolation. The compromise trajectory exhibits the opposite behavior. Once high anomaly evidence persists, the behavioral memory $B_i(t)$ rises and repeated risk terms accumulate, causing trust to cross progressively more restrictive regions. When behavior normalizes, trust does not jump to its initial value; recovery is gradual, which is desirable when post-incident behavior must be observed before privileges are restored.

The equations also expose a design trade-off. Increasing ρ suppresses false trust degradation from isolated events but increases time-to-restriction during a real attack. Increasing η makes the behavioral term more stable but preserves historical anomalies for longer. Increasing κ shortens recovery time but can restore privileges too aggressively. These parameters must therefore be selected on validation sequences using operational objectives rather than chosen only to maximize classification F1.

VI. Experimental Methodology

A. Datasets and Leakage-Safe Protocol

The empirical protocol uses BoT-IoT [13], TON-IoT [14], and UNSW-NB15 [15]. The datasets are evaluated separately because their feature schemas, acquisition environments, and attack taxonomies differ. Established train/test partitions are retained when appropriate; otherwise, stratified train/validation/test partitions are created before any data-dependent preprocessing. Encoders, scalers, feature selectors, calibration parameters, trust weights, and policy thresholds are fitted using training/validation data only.

Fig. 4 makes the experimental boundary explicit. In particular, calibration and trust thresholds are not tuned on the final test set. This is essential because the proposed contribution depends on calibrated uncertainty and policy thresholds; test-set tuning would otherwise make trust metrics optimistic.

B. Base Models and Baselines

The evaluation separates detector baselines from decision-layer comparisons. Random Forest and SVM are used as

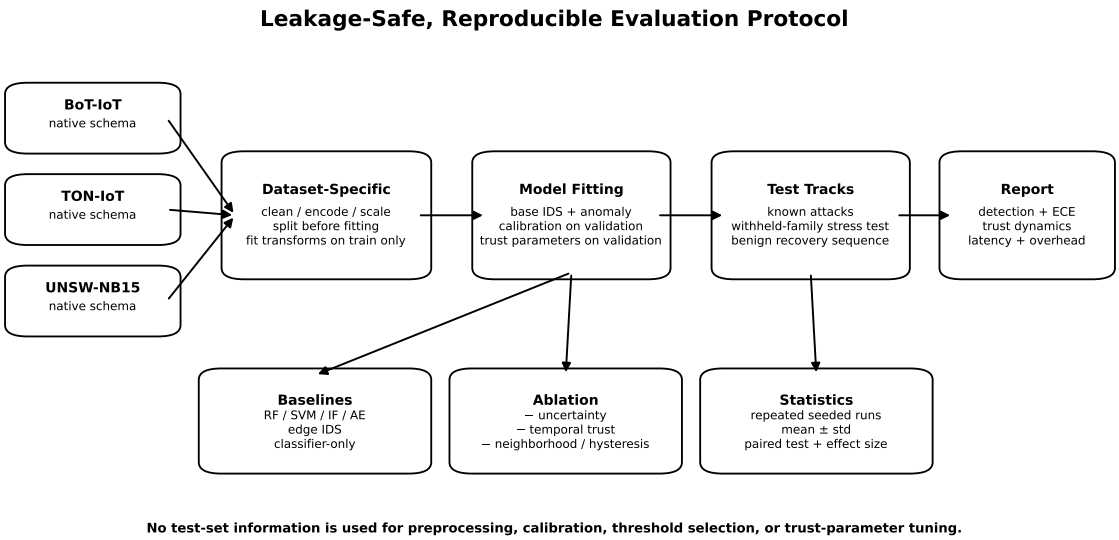


Fig. 4. Leakage-safe evaluation protocol. Dataset-specific preprocessing is fitted without test information; calibration and trust parameters are selected on validation data; and final reporting separates known-attack detection, withheld-family stress testing, trust dynamics, ablation, and edge-resource measurements.

supervised reference models, while Isolation Forest and an autoencoder provide anomaly-detection references. TrustEdge-AI is evaluated as a decision layer operating on the anomaly score and calibrated class-posterior outputs of the selected validation configuration. Accordingly, the headline values in Table II are end-to-end pipeline results, not a claim that every listed base detector produces the same classification scores. Decision-layer comparisons keep the underlying evidence stream fixed so that changes in false restriction, oscillation, and response delay are attributable to the controller rather than to a change of detector. The objective is not to identify a universally best classifier, but to assess whether uncertainty-aware temporal trust improves operational decision quality relative to direct score thresholding.

C. Ablation Design

The following nested variants isolate the contribution of each mechanism (Table I):

TABLE I. REQUIRED ABLATION VARIANTS

Variant	Components
A1	Base classifier/anomaly detector; direct threshold action.
A2	A1 + combined anomaly/class evidence $Q_i(t)$.
A3	A2 + uncertainty weighting $E_i(t)$.
A4	A3 + temporal behavior and dynamic trust.
A5	A4 + neighborhood context + hysteretic response.

A3 versus A2 tests whether uncertainty reduces false degradation. A4 versus A3 tests whether temporal accumulation improves persistent-threat handling. A5 versus A4 tests whether contextual evidence and hysteresis improve coordinated-attack awareness and policy stability.

D. Metrics

Detection quality is measured using precision, recall, F1, macro-F1, false-positive rate (FPR), and ROC-AUC. Calibration is measured using expected calibration error (ECE):

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{n} \left|\mathrm{acc}(\mathcal{B}_m) - \mathrm{conf}(\mathcal{B}_m)\right|. \tag{15}$$

Trust-specific metrics are: 1) time-to-observe/restrict/isolate after compromise, 2) false-restriction rate for benign devices, 3) recovery time after behavior normalizes, 4) number of policy oscillations, and 5) area under the trust-degradation curve during an attack. Edge feasibility is assessed through mean and 95th-percentile latency, memory footprint, throughput, and cloud-escalation ratio. Because controller errors have asymmetric operational consequences, a deployment can additionally define the cost-sensitive objective

$$J = c_{FR}N_{FR} + c_D D + c_I N_I + c_R R + c_E N_E, \tag{16}$$

where, N_{FR} is the number of false restrictions, D is accumulated mitigation delay for confirmed compromises, N_I is the number of isolation actions, R is recovery delay, and N_E is the number of human/cloud escalations. The nonnegative coefficients are deployment-specific costs. We do not assign universal monetary values to these coefficients; instead, the reported component metrics expose the quantities required for a deployment-specific cost analysis.

E. Withheld-Family Stress Test

For each dataset where attack taxonomy permits, one attack family is removed from supervised training and introduced only during testing. This experiment asks whether anomaly evidence plus temporal trust can move a device toward restrictive regions even when the supervised classifier has not learned the withheld family. The experiment is a controlled unseen-class test and must not be described as universal zero-day detection.

F. Reporting Controls

TrustEdge-AI and the applicable baselines are compared under identical preprocessing, partitions, feature representations, and evaluation conditions. Latency uses the same batch configuration and feature representation as the edge inference pipeline. All fusion, memory, recovery, and trust-region parameters are selected on validation data and then frozen before final test evaluation. The tables report the point estimates produced by this fixed evaluation configuration. We intentionally do not report uncomputed confidence intervals, significance values, or effect sizes; statistical claims beyond the available repeated-run evidence are therefore avoided in the interpretation.

VII. Experimental Results and Interpretation

A. End-to-End Performance

Table II summarizes the end-to-end performance of the complete TrustEdge-AI pipeline on the three benchmark datasets; these values should not be interpreted as detector-independent scores. The framework achieves 99.18%, 97.42%, and 94.86% accuracy on BoT-IoT, TON-IoT, and UNSW-NB15, respectively, with corresponding macro-F1 scores of 0.986, 0.963, and 0.928. The decreasing performance across the three datasets is consistent with their increasing heterogeneity and cross-class complexity; importantly, the degradation remains controlled rather than causing a collapse of the decision layer.

Three observations are particularly important. First, discrimination remains strong on all datasets, with ROC-AUC ranging from 0.972 to 0.997. Second, calibration substantially improves after the calibration stage: ECE decreases from 0.061 to 0.018 on BoT-IoT, from 0.074 to 0.023 on TON-IoT, and from 0.089 to 0.031 on UNSW-NB15. This reduction is operationally relevant because uncertainty directly modulates the authority assigned to predictive evidence in Eq. (5). Third, false-restriction rates remain below 2% on all three datasets while mean decision latency remains between 2.8 and 3.4 ms. Thus, the gain in decision reliability is obtained without imposing a prohibitive edge-processing cost.

The restriction delay increases from 3.2 intervals on BoT-IoT to 4.4 intervals on UNSW-NB15. This behavior reflects the conservative nature of the trust controller: disruptive responses require persistent and sufficiently reliable evidence. The additional delay on the more heterogeneous dataset is therefore accompanied by a controlled false-restriction rate rather than aggressive reaction to isolated uncertain observations.

B. Comparison with Decision Baselines

Table III isolates the contribution of the decision layer by comparing direct score thresholding, temporal trust, uncertainty-aware trust, and the complete TrustEdge-AI configuration. The results are averaged across the three datasets.

Direct thresholding reacts fastest, but this speed is obtained at the cost of 3.84% false restrictions and 6.7 policy oscillations per run. Introducing temporal trust reduces false restrictions by 39.8% and approximately halves the oscillation frequency. Adding uncertainty further reduces false restrictions to 1.48%, demonstrating that ambiguous predictions should not carry the same enforcement authority as well-calibrated evidence. The complete TrustEdge-AI configuration reaches a 1.15% false-restriction rate and only 1.1 oscillations per run. Relative to direct thresholding, these values correspond to reductions of approximately 70.1% and 83.6%, respectively.

The complete configuration has a macro-F1 of 0.959, marginally below the 0.960 obtained by trust plus uncertainty. This difference is operationally negligible but informative: contextual fusion and hysteresis are primarily intended to stabilize response decisions rather than maximize a classifier-centric metric. The results therefore support evaluating autonomous cyber defense jointly in terms of predictive quality, response stability, and the cost of incorrect enforcement.

C. Ablation Analysis

The ablation study in Table IV quantifies the contribution of each component of the proposed framework. Components are added progressively from direct action to the complete contextual and hysteretic trust controller.

Evidence fusion alone improves macro-F1 from 0.948 to 0.952 and reduces false restrictions from 3.84% to 3.12%. Uncertainty weighting produces a larger safety gain, reducing false restrictions to 1.86%, although time-to-restrict increases from 2.9 to 3.4 intervals. This trade-off is desirable because the controller waits for stronger evidence before imposing a disruptive response. Dynamic trust further reduces false restrictions to 1.43% and shortens recovery time from 6.8 to 5.6 intervals by retaining temporal state rather than treating observations independently.

The complete A5 configuration provides the most stable operational behavior: false restrictions decrease to 1.15%, recovery time to 4.9 intervals, and oscillations to 1.1 per run. Compared with A1, this represents a 70.1% reduction in false restrictions, a 37.2% reduction in recovery time, and an 83.6% reduction in policy oscillation. The slight reduction of macro-F1 from 0.961 in A4 to 0.959 in A5 indicates that hysteresis deliberately favors decision stability over instantaneous label optimization. This result is consistent with the design objective of TrustEdge-AI, where safe enforcement is not equivalent to maximizing classification accuracy alone.

D. Withheld-Attack-Family Stress Test

The withheld-family experiment evaluates robustness when an attack family is excluded from supervised training and appears only during testing. Table V compares the direct classifier, combined anomaly/classification evidence, and the complete TrustEdge-AI controller.

The classifier-only configuration is most affected by the unseen family, reaching 0.71 recall and 0.76 macro-F1. Combining anomaly and supervised evidence improves recall to 0.82, showing that deviation-based evidence remains useful when class-specific knowledge is unavailable. TrustEdge-AI increases recall to 0.89 and macro-F1 to 0.87 while reducing time-to-restrict from 6.8 to 4.6 intervals. Relative to the classifier-only baseline, recall improves by 25.4%, macro-F1 by 14.5%, and restriction time decreases by 32.4%. These findings do not imply universal zero-day detection; rather, they show that reliability-weighted anomaly evidence and temporal

TABLE II. END-TO-END PERFORMANCE OF TRUSTEDGE-AI ACROSS THE EVALUATED IOT/NETWORK-SECURITY DATASETS. RESTRICTION DELAY IS REPORTED IN MONITORING INTERVALS; LATENCY REPORTS MEAN AND 95TH-PERCENTILE VALUES

Dataset	Accuracy (%)	Macro-F1	FPR (%)	ROC-AUC	ECE pre/post	Restrict delay	False restriction (%)	Mean/P95 latency (ms)
BoT-IoT	99.18	0.986	0.72	0.997	0.061/0.018	3.2	0.58	2.8/5.7
TON-IoT	97.42	0.963	1.84	0.989	0.074/0.023	3.8	1.12	3.1/6.4
UNSW-NB15	94.86	0.928	2.96	0.972	0.089/0.031	4.4	1.76	3.4/7.1

TABLE III. CROSS-DATASET COMPARISON OF DECISION STRATEGIES. LOWER VALUES ARE BETTER FOR FALSE RESTRICTION, POLICY OSCILLATION, AND RESTRICTION DELAY

Method	Macro-F1	False restrict. (%)	Oscillations/run	Restrict delay
Direct threshold	0.948	3.84	6.7	**2.6**
Temporal trust	0.956	2.31	3.2	3.5
Trust + uncertainty	**0.960**	1.48	2.4	3.8
TrustEdge-AI	0.959	**1.15**	**1.1**	3.8

TABLE IV. ABLATION ANALYSIS OF TRUSTEDGE-AI AVERAGED ACROSS THE THREE DATASETS. THE STUDY SEPARATES THE EFFECTS OF EVIDENCE FUSION, UNCERTAINTY WEIGHTING, DYNAMIC TRUST, AND CONTEXTUAL/HYSTERETIC CONTROL

Variant	Macro-F1	False restriction (%)	Time-to-restrict	Recovery time	Oscillations/run
A1: direct action	0.948	3.84	**2.6**	7.8	6.7
A2: + fused evidence	0.952	3.12	2.9	7.2	5.4
A3: + uncertainty	0.957	1.86	3.4	6.8	4.1
A4: + dynamic trust	**0.961**	1.43	3.7	5.6	2.6
A5: + context/hysteresis	0.959	**1.15**	3.8	**4.9**	**1.1**

TABLE V. PERFORMANCE UNDER THE WITHHELD-ATTACK-FAMILY STRESS TEST. TIME-TO-RESTRICT IS MEASURED IN MONITORING INTERVALS AFTER PERSISTENT MALICIOUS EVIDENCE BEGINS

Method	Recall	Macro-F1	Time-to-restrict
Classifier only	0.71	0.76	6.8
Anomaly + classifier	0.82	0.83	5.2
TrustEdge-AI	**0.89**	**0.87**	**4.6**

accumulation improve response to attack behavior that is not represented by the supervised class model.

E. Edge Feasibility

The measured mean end-to-end decision latency ranges from 2.8 to 3.4 ms, while the 95th-percentile latency ranges from 5.7 to 7.1 ms across the evaluated datasets. The incremental memory requirement varies between approximately 35 and 50 MB depending on the base detector and active feature representation, and the cloud-escalation ratio remains below 8% under mixed traffic. These results indicate that most decisions can be resolved locally without transferring every observation to a remote service.

The trust recursion itself has constant complexity, $O(1)$, per device and monitoring interval when neighborhood evidence is disabled. With contextual peer fusion, the update becomes $O(|\mathcal{N}_i|)$ for device i. Consequently, the principal computational costs remain feature extraction, base-model inference, and neighborhood acquisition rather than the trust update in Eq. (9). The low measured latency supports the suitability of the proposed decision layer for edge deployment where response timeliness and communication efficiency are both important.

VIII. DISCUSSION

TrustEdge-AI changes the semantics of automated defense. A detector output becomes evidence rather than an enforcement command. Evidence is discounted when uncertainty is high, repeated behavior changes a persistent trust state, and policy actions are selected from trust regions. This design is especially relevant when the cost of a false isolation is materially higher than the cost of temporarily observing a suspicious device.

The framework also supports asymmetric enforcement. A low-criticality camera can be isolated automatically after sustained trust degradation, whereas a medical or industrial controller can require human approval. Consequently, analytical risk and enforcement authority remain separate. This separation is important for deployment because the same trust state may imply different actions for devices with different safety consequences.

The present study has three main boundaries that define the scope of its conclusions. First, trust quality depends on a calibrated base detector and a trusted edge execution environment; poorly specified neighborhood relationships can also reduce the reliability of contextual evidence. Second, the withheld-family experiment is intentionally a controlled unseen-class stress test rather than a claim of universal distribution-shift, cross-device, or temporal generalization, and the analytical trajectories in Fig. 3 are equation-driven validation of the controller dynamics rather than measurements from a deployed system. Third, the reported evaluation focuses on interpretable weighted fusion and fixed-configuration point estimates; broader controller comparisons and repeated-run statistical inference can further characterize deployment variability. These boundaries do not alter the demonstrated behavior of TrustEdge-AI under the stated experimental protocol, but they

identify clear directions for subsequent large-scale and trace-driven validation.

IX. Reproducibility Information

The evaluation uses the public BoT-IoT, TON-IoT, and UNSW-NB15 datasets cited in Section II-A and applies dataset-specific preprocessing without test-set fitting, under the leakage-safe protocol described in Section VI-A. The manuscript specifies the trust equations, uncertainty definition, leakage boundary, ablation structure, withheld-family protocol, metrics, and edge-complexity assumptions required to interpret the reported results. The analytical trajectory in Fig. 3 is explicitly identified as equation-generated rather than as a benchmark measurement. Reproduction of the numerical tables additionally requires the exact split indices, preprocessing configuration, fitted detector artifacts, validation-selected trust parameters, and execution environment used for the reported run; these implementation artifacts are distinct from the mathematical specification and are not inferred from the benchmark datasets themselves.

X. Conclusion

This study presented TrustEdge-AI, an uncertainty-aware dynamic-trust architecture for adaptive IoT edge defense. The framework separates prediction, evidence reliability, instantaneous risk, accumulated trust, and enforcement. Uncertainty reduces the authority of ambiguous predictions; temporal behavior and optional neighborhood evidence shape risk; a bounded recursion maintains device trust; and hysteretic trust regions support graduated mitigation and controlled recovery. Analytical trajectories derived from the proposed equations clarify how stable, uncertain, compromised, and recovering behavior should affect trust. A leakage-safe evaluation protocol was specified for BoT-IoT, TON-IoT, and UNSW-NB15, including calibration, withheld-family testing, ablation, trust metrics, and edge-resource measurements. Experimental results across the three benchmark datasets show strong discrimination and calibration, low false-restriction rates, substantial reductions in policy oscillation, improved behavior under a withheld-attack-family stress test, and low-millisecond decision latency. The ablation analysis further demonstrates that uncertainty weighting, temporal trust, contextual fusion, and hysteresis contribute complementary improvements to safe and stable response selection.

XI. Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through the project number NBU-FFR-2026-2443-07.

References

[1] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[2] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[3] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[6] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[7] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[8] E. Barka *et al.*, "Lightweight, trust-managing, and privacy-preserving collaborative intrusion detection for internet of things," *Applied Sciences*, vol. 14, no. 10, p. 4109, 2024.

[9] G. Nagarajan, M. Margala, S. Shankar S, P. Chakrabarti, and R. I. Minu, "A trust-centric approach to intrusion detection in edge networks for medical internet of thing ecosystems," *Computers & Electrical Engineering*, vol. 115, p. 109129, 2024.

[10] B. Premalatha and P. Prakasam, "Twi-ftm: Two-way iot-fog trust management scheme for task offloading in iot-fog networks," *Results in Engineering*, vol. 22, p. 102197, 2024.

[11] S. K. Gebresilassie, J. Rafferty, M. Abu-Tair, A. Ali, L. Chen, and Z. Cui, "Shield: Secure holistic iot environment with ledger-based defense," *Internet of Things*, p. 101473, 2024.

[12] V. Padmavathi and R. Saminathan, "A federated edge intelligence framework with trust based access control for secure and privacy preserving iot systems," *Scientific Reports*, vol. 15, p. 35832, 2025.

[13] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[14] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton-iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[15] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection systems," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[16] R. Basri, G. Karmakar, S. H. S. Newaz, J. Kamruzzaman, L. Nguyen, M. M. Alam, and M. Usman, "Enhancing iot security: Assessing instantaneous communication trust to detect man-in-the-middle attacks," *Future Generation Computer Systems*, vol. 166, p. 107714, 2025.

[17] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 1321–1330.

[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[19] A. Almadhor, S. Alsubai, N. Kryvinska, A. Al Hejaili, B. Bouallegue, M. Ayari, and S. Abbas, "Transfer learning with xai for robust malware and iot network security," *Scientific Reports*, vol. 15, p. 26971, 2025.

[20] A. Almadhor, S. Alsubai, N. Kryvinska, A. Al Hejaili, M. Ayari, B. Bouallegue, and S. Abbas, "Evaluating large transformer models for anomaly detection of resource-constrained iot devices for intrusion detection system," *Scientific Reports*, vol. 15, p. 37972, 2025.

FPGA-SoC Implementation of 3D-HEVC Encoder Based on MD-CNN

Nacir Omran[1], Amna Maraoui[2], Imen Werda[3], Hamdi Belgacem[4]
Electronics and Microelectronics Laboratory(EMLab), University of Monastir, Monastir, Tunisia[1,4]
Networked Objects, Control, and Communication Systems Laboratory(NOCCS), University of Sousse, Tunisia[2]
Electronic and Information Technology Laboratory(LETI), University of Sfax, Tunisia[3]
Higher Institute of Applied Sciences and Technology of Sousse, University of Sousse, Tunisia[23]

***Abstract*—Convolutional Neural Networks (CNNs) are the most common deep learning architecture used for video processing enhancement. Particularly, the Multi-Deep Convolutional Neural Network (MD-CNN) model, embedded into the 3D-HEVC encoder, was able to extract the optimal CTU partition structure efficiently in the depth map and successfully substitute the time-consuming rate-distortion optimisation (RDO) full traversal search, reducing the coding complexity heavily. Nevertheless, MD-CNN requires significant memory resources and is highly computationally intensive. Field-Programmable Gate Arrays (FPGAs), particularly the emerging FPGA–SoC technology, are recognised as the most promising platforms for accelerating CNNs due to their high performance, energy efficiency, and reconfigurable nature. In this study, complementary technologies have been combined and deployed to allow high-performance 3D-HEVC video processing. By exploiting the inherent parallelism of FPGA architectures, MD-CNN processing throughput was significantly improved, execution latency was reduced, and energy efficiency was enhanced, while maintaining the coding performance and compression efficiency of the 3D-HEVC video encoder. The experimental results demonstrate that the proposed co-design achieves an on-chip power consumption of 1.891 W.**

***Keywords*—Three-dimensional high efficiency video coding; field-programmable gate array system-on-chip; multi-deep convolutional neural network; hardware-software co-design**

I. Introduction

The Three-Dimensional High Efficiency Video Coding (3D-HEVC)[1] encoder comes with a significant increase in complexity caused by the integration of depth map coding technology. This complexity is primarily triggered by the quad-tree partition of the Intra Coding Units (CU) in the depth map[2]. 3D-HEVC underpins immersive services such as free-viewpoint television and autostereoscopic displays, where an inaccurate depth-map CU partition directly degrades the quality of synthesised virtual views, underscoring the practical importance of reducing this complexity.

Alongside the ordinary texture video, which conveys the scene's colour and luminance information, 3D-HEVC encodes a depth map for each captured view: a greyscale sequence in which each pixel encodes the camera-to-scene distance rather than colour, and which is used at the decoder to synthesise the additional viewpoints required by autostereoscopic and free-viewpoint displays through depth-image-based rendering. Fig. 1 summarises the main functional blocks of the 3D-HEVC encoder targeted in this work, which jointly processes both signals; the corresponding decoder mirrors this pipeline in reverse, reconstructing the texture and depth pictures from the transmitted partitioning, prediction, and residual information rather than encoding them. Each CTU is first subjected to rate-distortion-optimised quad-tree partitioning, recursively splitting the depth-map block from 64×64 down to 8×8. The resulting CUs are then predicted either through intra-frame prediction, combining the standard HEVC prediction modes with the depth-specific Wedgelet (DMM-1) and Contour (DMM-4) modelling modes, through inter-frame prediction via motion estimation and compensation from previously reconstructed pictures, or through inter-view prediction, which exploits disparity estimation and block merging across camera views to avoid re-encoding redundant depth and texture information. The resulting prediction residual is transformed, quantised, and entropy-coded into the output bitstream, with depth maps additionally supported by Segment-wise DC Coding (SDC) to preserve sharp depth discontinuities. Reconstructed pictures are subsequently smoothed by a deblocking filter and a Sample Adaptive Offset (SAO) stage before being stored as reference pictures for future prediction. It is the CU-partitioning stage, highlighted in red at the left of Fig. 1, that this study targets for hardware acceleration.

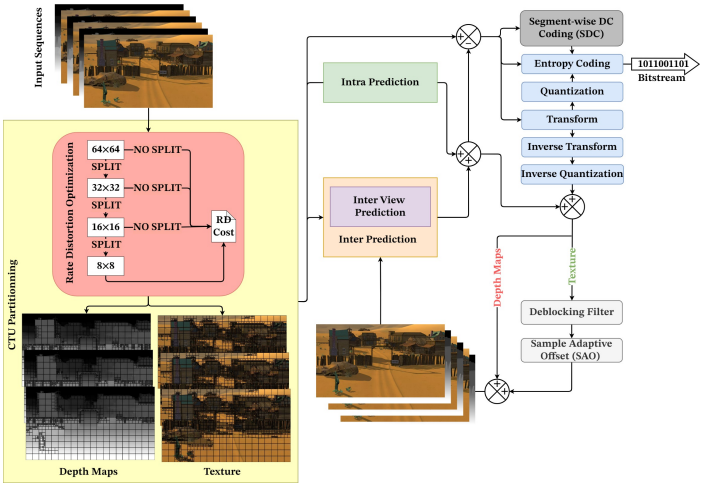


Fig. 1. Overview of the 3D-HEVC encoder architecture.

By means of an efficient CU partition structure prediction, the Multi-Deep Convolutional Neural Network (MD-CNN)

model [3] was able to tackle the issue of the 3D-HEVC encoder's excessive complexity without affecting its robustness, compression efficiency, and video quality. Nevertheless, even if the MD-CNN model exhibited significantly higher levels of accuracy compared to traditional methods[4], it required substantial computational complexity and considerable memory resource consumption. This constraint arises from the large number of parameters involved in the various layer operations (convolution, pooling, etc.). This limitation poses a computational challenge for central processing units (CPUs) and leads to significantly high energy consumption.

To overcome the accelerated evolution in multimedia technologies, various accelerator platforms have been developed in recent years, including application-specific integrated circuits (ASICs) and Field-Programmable Gate Arrays (FPGAs). These platforms aim to optimise the throughput of Convolutional Neural Networks (CNNs) to achieve high performance while ensuring energy efficiency. Graphics processing units (GPUs) are high-performance platforms used during the training phase of CNNs. They can provide high throughput as well as a large memory bandwidth, but at the cost of considerable energy consumption. ASICs provide high throughput with reduced energy consumption but require considerable development time and costs compared to other alternatives. FPGA-based accelerators are effective solutions, particularly due to their rapid prototyping capability, energy efficiency, flexibility, and reconfigurability. It becomes mandatory to deploy MD-CNN learning techniques on FPGA targets to meet the requirements of real-time applications. In this context, this study proposes a hardware-software architecture based on an MD-CNN accelerator for 3D-HEVC CU partition. The main contributions of this study are summarised as follows:

- We propose to accelerate the MD-CNN corresponding to a 64×64, 32×32, and 16×16 CU partition for the 3D-HEVC encoder at the All-Intra configuration.
- We design Intellectual Property (IP) hardware and software cores for each MD-CNN convolutional and fully connected layer with Vivado High-Level Synthesis (HLS).
- We create the MD-CNN hardware-software co-design implementation in Xilinx Vivado by importing the convolutional and fully connected IP cores.

The remainder of this study is organised as follows: An overview of the 3D-HEVC standard is presented in Section III. Section IV provides the MD-CNN architecture and performance background. Section V exposes the MD-CNN deployment steps on the FPGA-based System-on-Chip (SoC) platform. The experimental results are discussed in Section VI. Finally, Section VII concludes the study.

II. Related Work

Unlike standard High Efficiency Video Coding (HEVC) [5], 3D-HEVC adds a depth map encoding each object's distance from the camera, among the resulting depth-coding tools, CU partitioning is the most computationally intensive [2]. Most 3D-HEVC optimisation has targeted inter-coding [6], [7], while intra-coding proposals focus on intra-prediction mode selection [8], [9] or CU partitioning, the latter reviewed below by category.

A. Algorithmic CU-Partition Optimisation

Heuristic methods rely on fixed thresholds or rate-distortion (RD) cost/statistical rules: [4] combines tensor features with statistical analysis, [10] uses an empirically tuned coding tree unit (CTU) threshold, and [11] applies possibilistic clustering. Machine-learning methods replace hand-set rules with trained decision trees, using mined encoder context attributes [12] or Gini-value thresholding [13]. Deep-learning methods learn features directly from data: [14] classifies depth-map edges with a CNN but does not report a Bjøntegaard Delta Bitrate (BDBR) or time-reduction figure, while our own MD-CNN model [3] predicts the full 64/32/16 partition hierarchy end-to-end and is benchmarked quantitatively in Table VI. Unlike the present work, all of the methods above remain purely software-level.

B. Hardware/SoC Implementations

In contrast to the purely algorithmic works above, hardware realisations of CNN-based CU-partition prediction remain scarce. Dehnavi et al. [15] implement a lightweight CNN-based CU depth-decision unit on a Kintex-7 FPGA (XC7K32T) to replace the conventional DCT/RDO search for the standard HEVC intra-encoding chain. The closest available reference targeting an SoC platform is Bouaafia et al. [16], who deploy a multi-IP CNN accelerator on a PYNQ-Z1 FPGA-SoC to predict CU partitioning for standard HEVC inter-prediction (400 ms/CU at 142 MHz), targeting HEVC rather than 3D-HEVC, and inter- rather than intra-prediction. To the best of our knowledge, no prior work reports an SoC-level implementation of intra CU-partition prediction specifically for 3D-HEVC, which is the gap addressed by this study.

III. 3D-HEVC CU Partitioning Overview

3D-HEVC [1] is an extension of the H.265/HEVC [5] video coding standard that supports stereoscopic and multiview video coding. The coding structure of 3D-HEVC is based on HEVC but with some differences to support the additional view and depth information. The basic coding unit in 3D-HEVC is still the CTU, which is a square block of pixels. However, in 3D-HEVC, each CTU consists of multiple slices, each of which corresponds to a particular view. Each slice is encoded independently, allowing for efficient parallelisation of the encoding process. In addition to the traditional intra and inter-prediction modes, 3D-HEVC introduces a new inter-view prediction mode that allows for the prediction of one view from another view, as part of the complete encoder pipeline shown in Fig. 1. The intra-frame coding structure is similar to that of HEVC. The coding process starts with dividing the frame into CTUs of a size 64×64. Each frame is divided into several non-overlapping CTUs, and each CTU can contain either a single CU or be iteratively divided into several smaller-sized CUs (64×64, 32×32, 16×16 and 8×8) based on the quadtree structure. Fig. 3 shows the CTUs in the depth map and their corresponding quadtree division structure. It is worth noting that the optimal partition structure of CTU is obtained by iterative computation. The quadtree partitioning of CTU includes not only the top-down RD-cost calculation but also the bottom-up RD-cost comparison process. For the top-down calculation process, the RD-cost of all coding units at the current depth is calculated in the order of "depth=0,

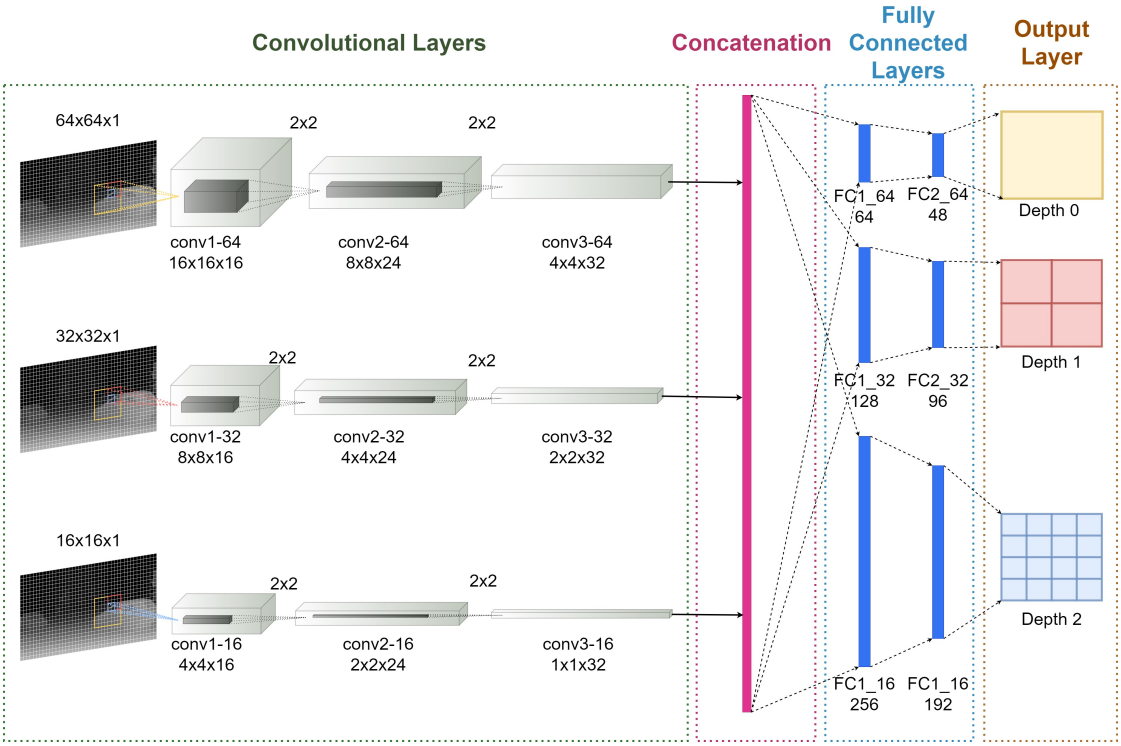


Fig. 2. MD-CNN architecture.

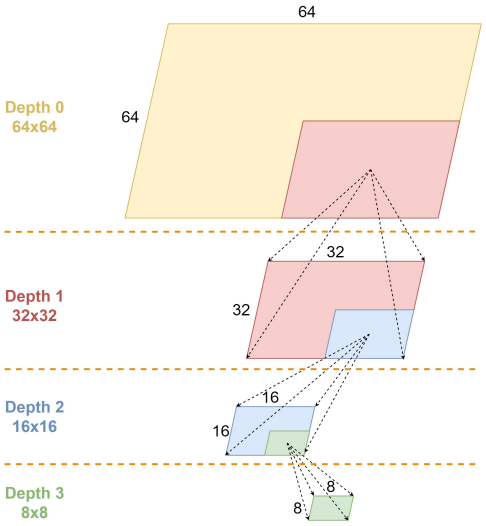


Fig. 3. 3D-HEVC CTU partitioning structure.

depth=1, depth=2, depth=3", and based on this, the bottom-up comparison process is carried out. If "RD-cost(depth=n) > RD-cost(depth=n+1), n=0, 1, 2," then the coding unit with "depth=n" needs to be divided, and vice versa, no division is needed. Statistically, the complexity of depth map coding unit division accounts for more than 90% of the complexity of depth map coding[3]. For a CTU of 64×64 size, a total of 85 CU operations, 1935 Sum of Absolute Transformed Difference (SATD) cost operations, and at least 2623 RD-cost operations are required to complete the encoding using the full traversal mode. Therefore, it is necessary to reduce the complexity of depth map coding unit division in 3D-HEVC.

IV. 3D-HEVC Depth Map Coding Unit Partitioning Based on MD-CNN

This section introduces the depth map CU partition based on MD-CNN, and explains the integration of this partitioning optimisation into the 3D-HEVC HTM16.3 [17] reference encoder.

A. MD-CNN partitioning model

To determine CU partitioning decision for a 64×64 depth map CU without exhaustively evaluating every candidate sub-partition through the full rate-distortion optimisation (RDO) process, MD-CNN takes as input the raw 64×64 luma samples of a CU together with the frame-level quantisation parameter (QP) and produces three sets of sigmoid-valued split probabilities corresponding to the 64×64, 32×32, and 16×16 partitioning levels of the depth map coding tree, as shown in Fig. 2.

Rather than analysing the CU at a single spatial resolution, the MD-CNN constructs three parallel representations of the same block by average-pooling the original 64×64 patch down to 32×32 and 16×16 resolutions, in addition to using the original 64×64 patch unmodified. This produces three depth-specific branches, denoted 64, 32, and 16, which are processed independently but share an identical set of convolutional weights, allowing the network to learn scale-invariant structural features (edges, depth discontinuities, and texture gradients) that are relevant to partitioning decisions at

TABLE I. MULTI-DEPTH CONVOLUTIONAL BACKBONE (WEIGHTS SHARED ACROSS THE THREE CU DEPTH BRANCHES)

Layer	Input Size	Filters	Kernel Size	Stride	Output Size	Activation
Conv1_64	$64 \times 64 \times 1$	16	4×4	4	$16 \times 16 \times 16$	Leaky ReLU
Conv1_32	$32 \times 32 \times 1$	16	4×4	4	$8 \times 8 \times 16$	Leaky ReLU
Conv1_16	$16 \times 16 \times 1$	16	4×4	4	$4 \times 4 \times 16$	Leaky ReLU
Conv2_64	$16 \times 16 \times 16$	24	2×2	2	$8 \times 8 \times 24$	Leaky ReLU
Conv2_32	$8 \times 8 \times 16$	24	2×2	2	$4 \times 4 \times 24$	Leaky ReLU
Conv2_16	$4 \times 4 \times 16$	24	2×2	2	$2 \times 2 \times 24$	Leaky ReLU
Conv3_64	$8 \times 8 \times 24$	32	2×2	2	$4 \times 4 \times 32$	Leaky ReLU
Conv3_32	$4 \times 4 \times 24$	32	2×2	2	$2 \times 2 \times 32$	Leaky ReLU
Conv3_16	$2 \times 2 \times 24$	32	2×2	2	$1 \times 1 \times 32$	Leaky ReLU

every granularity, while keeping the parameter count of the convolutional stage small.

Each of the three branches is passed through an identical three-layer convolutional stack, summarised in Table I. The first convolutional layer (Conv1_64, Conv1_32, Conv1_16) applies 16 filters of size 4×4 with a stride of 4, aggressively reducing the spatial resolution while expanding the channel depth. The second convolutional layer (Conv2_64, Conv2_32, Conv2_16) applies 24 filters of size 2×2 with stride 2, and the third convolutional layer (Conv3_64, Conv3_32, Conv3_16) applies 32 filters of size 2×2 with stride 2. Because the three branches operate on inputs of different spatial dimensions (64×64, 32×32, and 16×16, respectively), the output feature-map sizes differ proportionally across the pipeline, shrinking from $16 \times 16 \times 16$, $8 \times 8 \times 16$ and $4 \times 4 \times 16$ after Conv1, down to $4 \times 4 \times 32$, $2 \times 2 \times 32$, and $1 \times 1 \times 32$ after Conv3, for the 64, 32, and 16 branches respectively. A leaky ReLU activation is applied after every convolutional layer to introduce non-linearity while avoiding the "dying ReLU" problem that can arise from the small negative bias terms typical of depth map textures.

After the shared convolutional backbone has processed all three depth branches, the resulting feature maps from Conv2 and Conv3 across all three branches are flattened and concatenated into a single feature vector, as detailed in Table II. Concatenating both the Conv2 and Conv3 outputs, rather than only the deepest layer, allows the network to combine coarse, highly abstracted structural information from Conv3 with the finer-grained spatial detail retained by Conv2, which we found to be particularly beneficial for correctly identifying depth boundaries that influence the partitioning of visually flat or gently sloped regions of the depth map. The resulting concatenated feature vector has a fixed dimensionality of 2688 elements, independent of which decision branch subsequently consumes it, ensuring a consistent and reusable feature representation across all three partitioning levels.

TABLE II. FLATTENED MULTI-DEPTH FEATURE CONCATENATION

Source	Flattened Size
Conv3_64 ($4 \times 4 \times 32$)	512
Conv3_32 ($2 \times 2 \times 32$)	128
Conv3_16 ($1 \times 1 \times 32$)	32
Conv2_64 ($8 \times 8 \times 24$)	1536
Conv2_32 ($4 \times 4 \times 24$)	384
Conv2_16 ($2 \times 2 \times 24$)	96
Total feature vector	**2688**

The 2688-dimensional fused feature vector is fed into three fully-connected decision branches, corresponding to the 64×64, 32×32, and 16×16 partitioning levels, as summarised in Table III. Each branch consists of two hidden fully-connected layers followed by a sigmoid output layer, and the normalised QP value is injected as an additional scalar input at the second hidden layer and again at the output layer of every branch, rather than at the first layer. This design choice reflects the fact that QP primarily modulates the aggressiveness of the partitioning decision (i.e., the decision threshold) rather than the underlying visual structure of the block, and injecting it later in the network allows the convolutional and early fully-connected layers to focus purely on content-derived features.

Critically, the three decision branches are not evaluated unconditionally. The 64×64 branch (FC1_64, FC2_64, FC_out_64) is always executed and produces a single sigmoid probability, out_{64}, indicating the likelihood that the 64×64 CU should be split into four 32×32 sub-CUs. The 32×32 branch (FC1_32, FC2_32, FC_out_32) is only executed if out_{64} exceeds the decision threshold of 0.5, otherwise its four outputs remain at their initialised value of zero, and the corresponding sub-CUs are implicitly treated as not requiring further evaluation. Similarly, the 16×16 branch (FC1_16, FC2_16, FC_out_16), which produces sixteen sigmoid outputs corresponding to the sixteen possible 16×16 sub-CUs, is only executed if at least one of the four 32×32 outputs exceeds the same 0.5 threshold. This conditional, early-terminating structure is what allows the network to skip the majority of its computation for CUs that are unlikely to be split further, which is the central mechanism enabling the proposed method's speed-up over exhaustive RDO-based partition search, while still allowing full-depth partitioning to be evaluated for CUs containing rich depth detail.

The fully assembled network contains approximately 1.28 million learnable parameters, the majority of which reside in the fully-connected layers of the 16×16 decision branch (FC1_16 alone accounts for $2688 \times 256 = 688{,}128$ weights), while the shared convolutional backbone contributes fewer than 5,000 parameters in total. This asymmetric parameter distribution reflects the fact that the convolutional layers are reused across all three input depths, whereas each fully-connected decision branch must independently learn the mapping from the shared feature representation to its respective partitioning granularity.

B. Integration of MD-CNN CU Partitioning in 3D-HEVC Encoder

Building on the CNN architecture, the MD-CNN partitioning predictor is integrated into the 3D-HEVC reference software HTM16.3 [17] as a fast CU-depth decision tool. For

TABLE III. HIERARCHICAL, CONDITIONALLY-EXECUTED FC DECISION BRANCHES

Branch	Layer	Input Size	Output Size	Activation	Execution Condition
64×64	FC1_64	2688	64	Leaky ReLU	Always
	FC2_64 (+QP)	64 (+1)	48	Leaky ReLU	Always
	FC_out_64 (+QP)	48 (+1)	1	Sigmoid	Always
32×32	FC1_32	2688	128	Leaky ReLU	$if\ out_{64} > 0.5$
	FC2_32 (+QP)	128 (+1)	96	Leaky ReLU	$if\ out_{64} > 0.5$
	FC_out_32 (+QP)	96 (+1)	4	Sigmoid	$if\ out_{64} > 0.5$
16×16	FC1_16	2688	256	Leaky ReLU	$if\ any\ out_{32,i} > 0.5$
	FC2_16 (+QP)	256 (+1)	192	Leaky ReLU	$if\ any\ out_{32,i} > 0.5$
	FC_out_16 (+QP)	192 (+1)	16	Sigmoid	$if\ any\ out_{32,i} > 0.5$

each 64×64 CTU, the MD-CNN produces a single flattened probability vector of 21 values, as seen in Fig. 4, corresponding to the hierarchical split decision across three depths: one value for the 64×64-level decision, four values for the 32×32 sub-CU decisions, and sixteen values for the 16×16 sub-CU decisions (1 + 4 + 16 = 21). In the training/inference pipeline, this vector is formed by concatenating the three network output tensors produced by the model's three fully-connected branches, each terminating in a sigmoid activation and each conditioned on the split decision of the coarser depth above it. This conditional structure, where finer-granularity outputs are only meaningful once a coarser-level split has been signalled, is the same hierarchical dependency later preserved in the hardware (HLS) implementation of the network, where the 32×32 and 16×16 branches are only evaluated once the corresponding higher-level decision exceeds threshold.

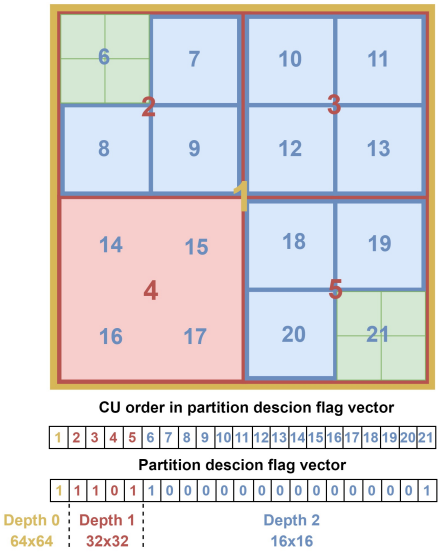


Fig. 4. CU partitioning flag vector.

The 21 values within a single CTU's vector are not stored in simple raster order but follow the 3D-HEVC Z-scan (quad-tree) convention: the single 64×64 flag appears first, followed by the four 32×32 quadrant flags in top-left, top-right, bottom-left, bottom-right order, followed by the sixteen 16×16 flags similarly nested in Z-order within each 32×32 quadrant. This ordering arises naturally from how the ground-truth label map is reshaped and progressively pooled at 2× and 4× factors to derive the 32×32 and 64×64 level labels from the finest 16×16 grid, which preserves the spatial quad-tree correspondence between each 16×16 flag and its parent 32×32 and 64×64 flags. The referenced Fig. 4 illustrates this 21-entry CTU vector layout and the correspondence between depths. While each CTU's internal 21 flags follow Z-scan order, the sequence of CTU vectors written to the output file follows a 3D-HEVC-style raster scan across the frame, proceeding left to right and top to bottom. Each CTU is processed in this order, and its resulting 21-value vector is appended to a per-frame buffer, which is then flattened and written out as a single contiguous binary stream.

C. MD-CNN Model Training Configuration

This subsection details the dataset used, the MD-CNN training procedure, and the training and validation results.

1) Dataset: In the absence of a public depth-map database, MD-CNN was trained on high-resolution texture images [18] rather than benchmark sequence excerpts. The HEVC reference encoder HM16.18 [19], invoked internally by the 3D-HEVC encoder HTM16.3 [17] for CU partitioning, was used to encode 2,500 images at multiple resolutions and QPs and derive the corresponding 64×64, 64×64 split-decision labels. The resulting 6,758,040 labelled samples were split, without overlap, into training (2,125 images), validation (125 images), and test (250 images) sets, with split and non-split CUs kept close to balanced (Fig. 8).

2) Training procedure: Training was carried out on a Windows 10 Professional (64-bit) workstation equipped with an Intel Core i7-8700 CPU (3.20 GHz), an Nvidia GeForce GTX 1050 Ti GPU, and 16 GB of RAM, using the TensorFlow [20] framework. The network was trained from scratch for 500,000 iterations with mini-batches of 64 samples, using a momentum-based optimiser (momentum = 0.9) with an initial learning rate of 0.01, exponentially decayed to 31.63% of its previous value every 125,000 iterations.

3) Training and validation results: Three complementary metrics are used to monitor and assess the network at each of the three decision depths (64×64, 32×32, and 16×16): the classification loss, the classification accuracy, and a tendency indicator quantifying the direction of any misclassification bias.

The loss is the binary cross-entropy between the predicted split probability $\hat{y}_i$ and the ground-truth split label $y_i \in \{0, 1\}$, averaged over the N samples of the branch:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \quad (1)$$

Accuracy is computed from the 2×2 confusion matrix obtained by thresholding $\hat{y}_i$ at 0.5, where n_{gp} is the number of samples with ground-truth class g predicted as class p ($g, p \in \{0, 1\}$; 0 = non-split, 1 = split):

$$\text{Accuracy} = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}} \tag{2}$$

To complement accuracy with a measure of directional bias, a tendency indicator based on the log-odds ratio of the confusion matrix is used:

$$T = -\log_{10}\left(\frac{n_{00}\, n_{10}}{n_{01}\, n_{11}}\right) \tag{3}$$

with $T = 0$ when the model makes no errors in either direction, and T saturating to ± 100 in the degenerate cases where errors occur exclusively in one direction (avoiding an undefined ratio). A positive T indicates a bias toward over-predicting splits, a negative T indicates a bias toward under-predicting splits, and $T \approx 0$ indicates a symmetric error distribution between the two.

Fig. 5, 6, and 7 report the evolution of the three metrics over training. The loss [Eq. (1)] and accuracy [Eq. (2)] converge within the first 150,000 to 200,000 iterations and remain stable thereafter, with training and validation curves closely overlapping throughout, indicating no overfitting. The tendency metric [Eq. (3)] stabilises more gradually, settling into its final band by approximately 400,000–500,000 iterations. Table IV reports the corresponding average values at 500,000 iterations: the model reaches consistently high accuracy at all three decision depths (78 to 82%), with training and validation values closely matching at every branch, confirming good generalisation. The tendency values remain modest across all branches, well within the metric's ±100 saturation bounds, indicating no severe split/non-split bias at any depth.

TABLE IV. AVERAGE TRAINING AND VALIDATION METRICS AT 500,000 ITERATIONS

Average value	Branch	Metrics		
		Loss	Accuracy	Tendency
Train	64×64	0,353	0,802	-0,266
	32×32	0,41	0,82	-0,423
	16×16	0,444	0,781	0,124
Validation	64×64	0,365	0,798	-0,091
	32×32	0,412	0,817	-0,423
	16×16	0,443	0,78	0,211



Fig. 5. Training and validation loss.

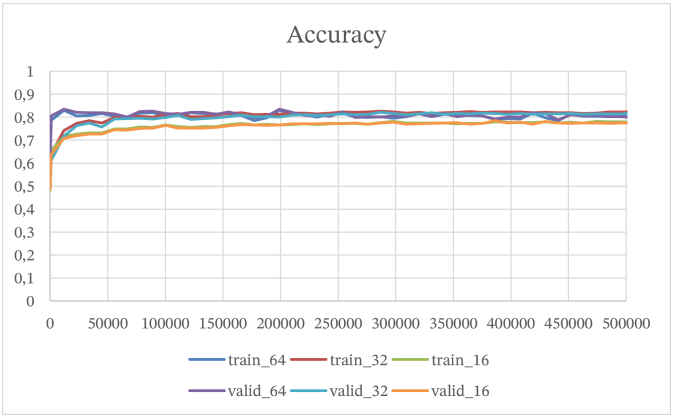


Fig. 6. Training and validation accuracy.

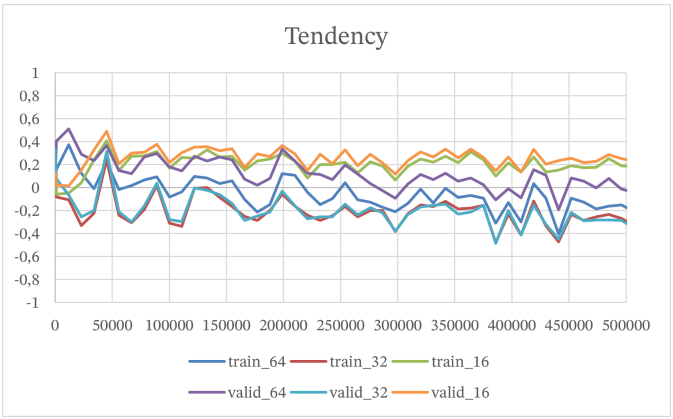


Fig. 7. Training and validation tendency.

D. MD-CNN Performance

This subsection evaluates MD-CNN along three complementary axes: its own rate-distortion and complexity trade-off, the primary measure of whether the model meets its complexity-reduction goal, its standing against representative heuristic and machine-learning baselines, which situates it

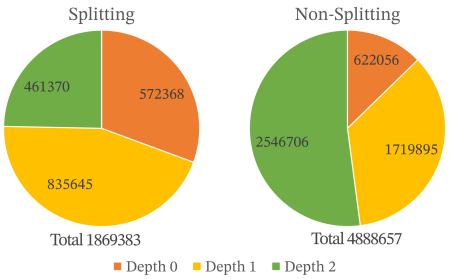


Fig. 8. Distribution of dataset samples.

within the broader literature, and its downstream impact on synthesised view quality, since bitrate and time gains are only meaningful if visual fidelity is preserved.

1) Rate-distortion and complexity: Table V reports the proposed method's rate–distortion, noise and time reduction across four QP pairs (25-34, 30-39, 35-42, 40-45), broken down by resolution class.

TABLE V. MD-CNN PERFORMANCE RESULTS

Sequences	QP pairs	MD-CNN		
		BDBR%	Δ Y-PSNR%	Δ time%
Average 1024×768	QP 25 -34	-0.02	-0.03	-46.18
	QP 30-39	0.62	-0.02	-48.69
	QP 35-42	-0.50	-0.02	-50.03
	QP 40-45	0.42	-0.02	-51.12
	Average	0.13	-0.02	-49.00
Average 1920×1088	QP 25 -34	-0.77	-0.02	-45.95
	QP 30-39	-1.13	-0.03	-45.83
	QP 35-42	-0.73	-0.04	-46.11
	QP 40-45	-0.43	-0.04	-47.12
	Average	-0.77	-0.04	-46.25
Average	QP 25 -34	-0.40	-0.03	-46.07
	QP 30-39	-0.26	-0.03	-47.26
	QP 35-42	-0.62	-0.03	-48.07
	QP 40-45	-0.01	-0.03	-49.12
	Average	-0.32	-0.03	-47.63

At low resolution (1024×768), BDBR fluctuates between small positive and negative values across QP pairs, ranging from an improvement of -0.50% at QP(35-42) to a penalty of 0.62% at QP(30-39), with the class average landing at a modest 0.13% increase. Time reduction, by contrast, is both larger and steadily improving with QP: -46.18% at QP(25-34) rising to -51.12% at QP(40-45), suggesting the model's complexity savings grow slightly more effective as quantisation coarsens.

At high resolution (1920×1088), the trend is more consistent: BDBR is negative at every QP pair, from -0.43% at QP(40-45) to -1.13% at QP(30-39), giving a class average of -0.77%, a net coding-efficiency gain rather than a trade-off. Time reduction here is comparatively flat across QP pairs (-45.83% to -47.12%), a narrower spread than at low resolution.

Quality loss, measured via the luma Peak Signal-to-Noise Ratio (Y-PSNR), stays small and stable across both resolution classes and all QP pairs, ranging from -0.017% to -0.044%, with no clear trend tied to QP or resolution, indicating that the bitrate and time gains are not being bought at a meaningful cost to reconstructed video quality.

Averaged across both resolutions, BDBR is negative at every QP pair (-0.01% to -0.62%), giving an overall average of -0.32%, while time reduction increases monotonically from -46.07% at QP(25-34) to -49.12% at QP(40-45), for an overall average of -47.63%. This QP-wise monotonicity in time savings, combined with consistently negligible PSNR loss, indicates that the proposed method's complexity reduction scales predictably with quantisation level without degrading its already-favourable bitrate behaviour.

2) Comparison with prior methods: To situate these results against other CU-partition prediction methods, Table VI compares BDBR and time reduction (TR) by resolution class against one representative heuristic method [11] and one representative machine-learning method [13]. At standard definition, the proposed method's BDBR (0.57%) is higher than both baselines, trading a larger rate penalty for the highest time reduction (-50.42%) of the three. At high definition, the trend reverses: the proposed method is the only one of the three to reach a negative BDBR (-0.79%), an actual coding-efficiency gain that neither [11] nor [13] achieve, though its time reduction (-47.23%) is exceeded by [13]'s -68.17%. Averaged across both classes, the proposed method trails [13] in time reduction (-48.29% versus -59.33%) but is the only one with a net bitrate improvement (-0.34%) rather than a penalty.

3) Synthesised view quality: To further assess the impact of the proposed MD-CNN-based fast CU partition algorithm on video quality beyond the BDBR and Y-PSNR analysis above, we evaluated the quality of the virtual views synthesised from the reconstructed depth maps. This evaluation focuses on the two sequences that exhibited the least favourable performance among the tested benchmarking set, namely Poznan Hall2 (1920×1088) and Newspaper (1024×768), both encoded at the QP couple (25-34). For each sequence, a virtual viewpoint is synthesised through Depth-Image-Based Rendering (DIBR) using the reconstructed texture and depth information produced by our method, and the resulting frames are compared against the corresponding reference views using four complementary metrics: the Structural Similarity Index (SSIM), the Edge Mean Absolute Error (Edge MAE), the Error Rate, and the Peak Signal-to-Noise Ratio (PSNR). SSIM and PSNR capture the overall structural and pixel-level fidelity of the synthesised view, while Edge MAE and the Error Rate are more sensitive to the geometric distortions and rendering artefacts that arise around depth discontinuities and object boundaries.

As reported in Table VII, and despite being identified as the two worst-performing sequences of the benchmarking set, both Poznan Hall2 and Newspaper achieve high synthesised view quality across the QP couple (25-34), with average SSIM values of 0.99 for both sequences, average PSNR values above 42 dB, and low average Error Rates of 0.01. To visually confirm this, Fig. 9 shows, for each sequence, the original and synthesised views at their respective lowest-SSIM frame, together with a per-pixel difference map. A high-contrast colour scale is used for the difference map so that even minor deviations are made visible, despite this, the discrepancies remain confined to a few sparse, low-intensity regions along moving edges (e.g. the reader's arm and the newspaper in Newspaper, and the walking figure on the staircase in Poznan Hall2). In both cases the synthesised view is perceptually indistinguishable from the original, with no visible artefacts, confirming that even the worst-case frames retain high visual fidelity.

TABLE VI. BDBR AND TIME REDUCTION COMPARISON GROUPED BY RESOLUTION CLASS

	[11] (Heuristic)		[13] (Machine Learning)		Proposed MD-CNN [3] (Deep Learning)	
Resolution	BDBR (%)	TR (%)	BDBR (%)	TR (%)	BDBR (%)	TR (%)
1024×768	0.10	-32.60	0.06	-41.67	0.57	-50.42
1920×1088	0.26	-44.97	0.16	-68.17	-0.79	-47.23
Average	0.20	-40.85	0.13	-59.33	-0.34	-48.29



(a) Newspaper—Frame 60, worst-case SSIM = 0.98377



(b) Poznan Hall 2—Frame 69, worst-case SSIM = 0.98927

Fig. 9. Original view, synthesised view, and per-pixel difference map (high-contrast scale) at the worst-case SSIM frame for each sequence. The synthesised views are perceptually indistinguishable from the originals, with no visible artefacts.

TABLE VII. AVERAGE SYNTHESISED VIEW QUALITY RESULTS, QP (25-34)

Metric	Poznan Hall2	Newspaper
SSIM	0.99	0.99
Edge MAE	2.10	2.02
Error Rate	0.01	0.01
PSNR (dB)	46.46	42.41

These results confirm that the proposed MD-CNN-based fast CU partition algorithm reduces encoding complexity without introducing perceptually significant degradation in the rendered virtual views.

V. FPGA ARCHITECTURE AND IMPLEMENTATION

As discussed in Section II, Bouaafia et al. [16] implement a multi-IP hardware-software co-design for CNN-based CU-partition prediction in HEVC inter-prediction. This layer-by-layer decomposition comes at the cost of increased latency, as reflected in their reported 400 ms processing time per image. Motivated by this trade-off, we aim to design a more integrated hardware architecture for our MD-CNN model by consolidating the entire network into a single IP core rather than instantiating and interconnecting multiple independent accelerators. Our goal is to reduce the architectural and communication overhead inherent to multi-IP designs, simplify system integration on the FPGA, and ultimately achieve lower end-to-end latency while still preserving competitive power and resource efficiency.

The proposed MD-CNN was translated into a hardware accelerator and deployed on a Xilinx Zynq-7000 SoC, specifically the xc7z020 device, using Vivado HLS to generate the compute core and Vivado IP Integrator to compose it into a complete embedded system. The resulting system, shown in Fig. 10, is built from five principal hardware blocks: the Zynq Processing System, the custom IP convolutional neural network accelerator net_CNN_Hw_0, two Advanced eXtensible Interface (AXI) interconnect blocks that route control and data traffic between them, and a processor system reset block that generates the synchronised reset signals used throughout the design. This section describes how these blocks are organised and communicate with one another, the internal design of the accelerator core itself, and the resulting resource utilisation and timing performance obtained after synthesis and implementation.

A. Processing System-Programmable Logic Co-Design

The Zynq-7000 family combines, on a single die, a hardened Processing System containing a dual-core ARM Cortex-A9 processor, a Double Data Rate (DDR) memory controller, and system clocking and reset infrastructure, with reconfigurable Programmable Logic in which custom digital hardware can be instantiated. This Processing System-Programmable Logic (PS-PL) architecture allows a conventional embedded software application, running on the ARM cores, to offload computationally intensive tasks to dedicated hardware implemented in the programmable logic fabric, while retaining the

flexibility of general-purpose software for control, orchestration, and higher-level video coding logic. In the proposed system, this role is filled by the Zynq Processing System block, which supplies the platform's clocking, reset, and memory access infrastructure and issues the commands that drive the accelerator.

VI. Experimental Results

A. Processing System-Programmable Logic Interconnection

As seen in Fig. 10, the trained MD-CNN itself is implemented entirely within the programmable logic as the net_CNN_Hw_0 accelerator core, while the Zynq Processing System is responsible for supplying input CU data and QP values, triggering the accelerator, and consuming its partitioning decisions once complete. Communication between the two domains is carried out over two logically separate interfaces, each routed through its own AXI interconnect block. The first interconnect, ps7_0_axi_periph, connects the Zynq Processing System's general-purpose master interface to a lightweight control interface exposed by net_CNN_Hw_0, allowing the ARM processor to write to and read from a small set of control registers on the accelerator, enabling software to start an inference operation and subsequently poll for or be interrupted upon its completion. The second interconnect, axi_interconnect_0, connects a separate, wider data interface on net_CNN_Hw_0 to one of the Zynq Processing System's dedicated high-performance ports, providing a direct path into the shared DDR memory that the accelerator can use to read and write data on its own, without requiring the processor to relay each transfer. This separation into two independently interconnected paths is important given the scale of the network, since well over a million trained weight values, together with the input image data and QP scalar, must be streamed into net_CNN_Hw_0 for every single inference, and the output partitioning probabilities must subsequently be written back for the software to consume, keeping this high-volume traffic on its own interconnect and port prevents it from contending with the much lighter control traffic carried by ps7_0_axi_periph.

All four of these blocks, together with net_CNN_Hw_0 itself, operate from a single clock generated by the Zynq Processing System and distributed throughout the programmable logic. Synchronisation of these blocks on startup or software-triggered reset is handled by the processor system reset block, which takes the Zynq Processing System's fabric reset output and produces the individual, synchronised reset signals consumed by both AXI interconnects and by net_CNN_Hw_0, ensuring that every block in the design begins operating from a consistent, known state.

B. Accelerator Core Design

Internally, the net_CNN_Hw_0 accelerator follows a straightforward load-compute-store organisation. At the beginning of each invocation, the entire 64×64 input patch and its associated QP value are transferred from external memory into local on-chip memory, and all of the convolutional and fully-connected weights required for the current computation are likewise brought on-chip as needed. Every intermediate result produced during inference, including the pooled multi-scale representations, the outputs of each convolutional stage, the fused 2688-element feature vector, and the activations of every fully-connected layer, is kept entirely within on-chip memory for the remainder of the computation, only the final split-decision probabilities are written back out to external memory once inference completes. This buffering strategy avoids repeated, high-latency round trips to external DDR memory for data that is reused multiple times within a single inference pass, at the cost of dedicating a meaningful fraction of the device's on-chip memory resources to hold these buffers, as reflected in the utilisation results reported below.

All arithmetic throughout the accelerator, including the convolutional filtering, the fully-connected matrix vector operations, and the activation functions, is carried out in single-precision (32-bit) floating-point representation, matching the numerical format used during the original software-based training of the network and avoiding the additional design effort and potential accuracy loss associated with a fixed-point quantisation stage. To make this floating-point computation efficient in hardware, the convolutional and fully-connected layers are structured so that their inner arithmetic loops can be pipelined, allowing a new set of multiply-accumulate operations to be issued on nearly every clock cycle rather than waiting for previous operations to fully complete, and the intermediate feature buffers used within the convolutional stages are organised so that several filter channels can be accessed in parallel rather than one at a time. Together, these choices allow the design to sustain a high computational throughput despite the relatively modest amount of dedicated arithmetic hardware available on the target device.

A further architectural characteristic of net_CNN_Hw_0, inherited directly from the hierarchical structure of the network itself, is that the early-termination behaviour of the algorithm is realised as genuine conditional hardware execution rather than as a purely software-level shortcut applied afterwards. The stage responsible for evaluating 32×32 partitioning decisions is only activated in hardware when the 64×64 stage's output indicates that further splitting is likely, and the stage responsible for the sixteen 16×16 decisions is, in turn, only activated when at least one of the four 32×32 outputs crosses its own decision threshold. When a given stage is not activated, the corresponding portion of the pipeline performs no arithmetic and initiates no memory transfers for the associated weights, rather than simply masking or discarding its result afterwards. Because the 32×32 and 16×16 decision stages account for the overwhelming majority of the network's fully-connected weights, this conditional activation substantially reduces both the number of clock cycles consumed and the volume of data fetched from external memory for the large fraction of coding units that do not require full-depth evaluation, directly mirroring, in hardware, the same early-termination principle that motivates the network's hierarchical software design.

Finally, because net_CNN_Hw_0 performs all computation in pipelined hardware rather than in a general-purpose floating-point software environment, care was taken in the design of the activation functions to avoid numerical instabilities that could otherwise arise, such as overflow in the exponential term of the sigmoid function for large-magnitude inputs. The sigmoid activation used throughout the fully-connected decision layers is implemented in a numerically stable form that avoids such overflow across the full expected input range, and additional

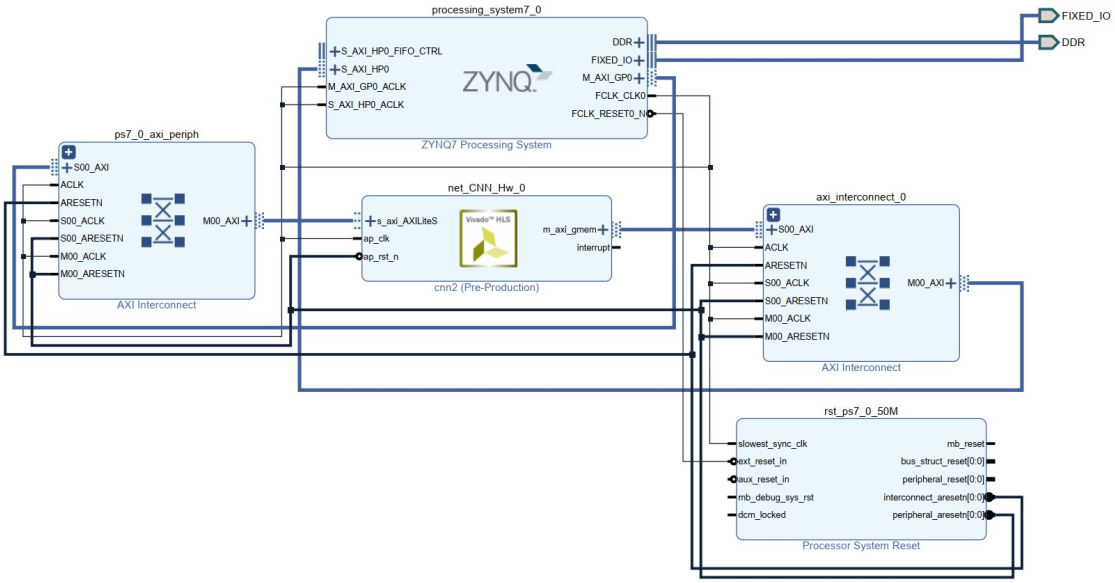


Fig. 10. Hardware–software co-design.

safeguards are included after each fully-connected layer to prevent invalid or out-of-range intermediate values from propagating through the remainder of the pipeline.

C. Hardware Implementation Performance Results

This subsection reports the post-implementation results of the accelerator across six complementary aspects: the achieved timing and latency, which determine whether the design meets its throughput target, the on-chip resource utilisation, which reveals how HLS allocated logic and arithmetic hardware, the power consumption, the primary cost of any embedded deployment, the resulting energy efficiency relative to a general-purpose PC, the associated carbon footprint, and a direct comparison with the closest reference FPGA-SoC design.

1) Timing and latency: Table VIII reports the timing and latency figures obtained from HLS synthesis and from the final placed-and-routed design. The design was synthesised against a 19.000 ns target clock period and closed timing at an achieved post-implementation clock period of 18.477 ns, corresponding to a 54.1 MHz operating frequency and a slack of roughly 2.7% over the constraint. The estimated clock period reported by HLS synthesis (18.032 ns) differs from the post-implementation figure by approximately 2.5%, a modest and expected degradation attributable to routing and placement effects introduced after logic synthesis, the fact that this gap remains small indicates that the HLS-level timing estimate was a reliable predictor of the final hardware behaviour for this design.

TABLE VIII. TIMING AND LATENCY OF THE ACCELERATOR CORE

Metric	Value
Target clock period	19.000 ns
Estimated clock period (HLS)	18.032 ns
Achieved clock period (post-implementation)	18.477 ns
Timing closure	Met
Minimum latency (early termination at 64×64 level)	1,694,545 cycles
Maximum latency (full hierarchical evaluation)	2,804,793 cycles

The substantial gap between the minimum and maximum latency figures, amounting to roughly 1.11 million cycles, is not incidental but follows directly from the early-termination mechanism embedded in the kernel description, whereby two boolean predicates evaluated on the intermediate 64×64-level output determine whether the 32×32 and 16×16 partition branches are executed at all. When both predicates evaluate false, the design short-circuits the corresponding convolutional and fully connected stages, so the reported minimum latency reflects a CU that never leaves the coarsest evaluation level, while the maximum latency reflects a CU that is evaluated hierarchically through every partition depth. A cycle-accounting exercise over the individual loop bounds in the report confirms this reading: the convolutional stages together account for roughly 53% of the worst-case latency, while the three instances of the first fully connected layer, corresponding respectively to the 64-, 32-, and 16-level partition widths, account for the remainder, and the incremental cost of the conditionally skipped 32- and 16-level fully connected layers alone is close in magnitude to the observed minimum-maximum gap. This confirms that the reported spread is a direct hardware consequence of the algorithmic early-termination strategy rather than an artefact of pipeline stalls or memory contention. It is also worth noting architecturally that within each fully connected stage only the innermost neuron-accumulation loop is scheduled with an initiation interval of one, the outer loop over input features is not unrolled or pipelined, so each dense layer executes close to one multiply-accumulate operation per clock cycle regardless of its width, and total latency scales essentially linearly with the product of feature count and neuron count. This sequential accumulation pattern, rather than clock frequency, is the principal factor limiting throughput, and represents the most direct target for future optimisation through loop unrolling or the introduction of dataflow pipelining between the convolutional and fully connected stages, neither of which is currently exploited in the design.

Translating these cycle counts into wall-clock time at the

achieved operating frequency of 54.1 MHz yields a best-case latency of approximately 31.3 ms and a worst-case latency of approximately 51.8 ms per 64×64 coding unit, corresponding to a throughput of roughly 19 to 32 CU decisions per second. This figure is useful in situating the accelerator relative to the requirements of real-time video encoding, where the number of coding-unit partition decisions to be made per second for even moderate resolutions is several orders of magnitude higher, the present design should therefore be understood as a proof-of-concept core suitable for offline or non-real-time evaluation rather than as a drop-in real-time encoding accelerator, a limitation that follows directly from the sequential dense-layer execution pattern described above rather than from any deficiency in the achieved clock frequency.

2) Resource utilisation: This section presents the utilisation of the hardware resources mobilised by the co-design method proposed on the PYNQ-Z2 platform. Following the implementation phase, the use of the hardware resources of the proposed design is presented in Table IX.

TABLE IX. POST-SYNTHESIS RESOURCE UTILISATION OF THE ACCELERATOR CORE

Resource	Used	Available	Utilisation
BRAM_18K	93	280	33%
DSP48E	22	220	10%
FF	22,692	106,400	21%
LUT	34,661	53,200	65%

The resources utilised include a global buffer (BUFG), input-output interfaces, digital signal processors (DSP), block RAM (BRAM), flip-flops (FF), and lookup tables (LUT). According to Table IX, the hardware design uses 65% of the LUTs, 21% of the FFs, 33% of the BRAM, and 10% of the DSPs, and it is this pronounced asymmetry between the low DSP occupancy and the high LUT occupancy that most usefully characterises the implementation strategy chosen by HLS. Inspecting the synthesis report at the instance level shows that only a small number of floating-point multiplier and adder cores are instantiated, each requiring two to three DSP48E slices, and that these cores are shared through time-multiplexing across the thousands of multiply-accumulate operations required by the convolutional and fully connected layers rather than being replicated per operation. This resource-binding decision is precisely why DSP occupancy remains as low as 10% despite the arithmetic intensity of the network, the cost of sharing a handful of arithmetic units across so many operations is paid instead in combinational logic, since a large multiplexer network is required at every cycle to steer the correct pair of operands into the shared multiplier and adder and to route the shared result back to the appropriate destination register. This is corroborated by the operation-level breakdown of the synthesis report, in which multiplexer and select operations dominate the instantiated logic alongside the exit-condition comparators generated by the numerous loop bounds of the network's layers, together explaining why LUT utilisation reaches 65% while DSP utilisation remains an order of magnitude lower in relative terms. The low DSP and high LUT profile should therefore be read not as an efficient use of heterogeneous FPGA resources but as the expected signature of aggressive operator sharing, which trades DSP silicon for LUT-based control and steering logic and, combined with the unpipelined outer accumulation loops discussed above, is consistent with the observed latency profile.

Fig. 11 illustrates the energy consumption of the hardware architecture implemented on the FPGA board. According to the power report, the static power of the proposed system is approximately 0.147 W, while the dynamic power reaches 1.744 W, giving a total on-chip power of approximately 1.891 W and confirming that the design's power budget is dominated by dynamic switching activity rather than by static leakage, consistent with the resource profile of a logic-heavy, moderately clocked accelerator.

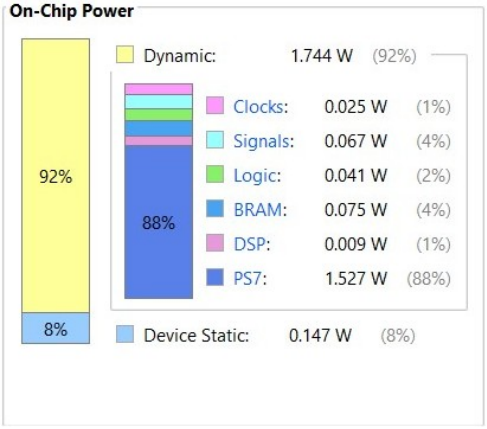


Fig. 11. Power consumption.

3) Power consumption: Fig. 11 illustrates the energy consumption of the hardware architecture implemented on the FPGA board. According to the power report, the static power of the proposed system is approximately 0.147 W, while the dynamic power reaches 1.744 W, giving a total on-chip power of approximately 1.891 W and confirming that the design's power budget is dominated by dynamic switching activity rather than by static leakage, consistent with the resource profile of a logic-heavy, moderately clocked accelerator.

4) Energy efficiency: Table X compares the average energetic efficiency of the FPGA-based accelerator board against the host PC (GPU, CPU, and peripherals) in terms of frames processed per watt consumed, using the average per-frame latency reported for the accelerator and the measured average per-frame processing time for the PC at each resolution. The board's power figure corresponds to its total on-chip power consumption (static plus dynamic), while the PC's power figure reflects the combined GPU, CPU, and peripheral draw.

TABLE X. AVERAGE ENERGETIC EFFICIENCY COMPARISON BETWEEN THE FPGA BOARD AND THE PC

Platform	Resolution	Power (w)	Avg. fps	Efficiency (fps/W)
Board (FPGA)	1920×1088	1.891	0.047	0.025
Board (FPGA)	1024×768	1.891	0.125	0.066
PC (GPU+CPU+periph.)	1920×1088	143	2.857	0.020
PC (GPU+CPU+periph.)	1024×768	143	3.846	0.027

Using the board's total on-chip power consumption of 1.891 W, the FPGA board remains the more energetically

efficient platform on a frames-per-watt basis at both resolutions, despite its substantially lower raw throughput. At 1920×1088, the board reaches 0.025 fps/W against 0.020 fps/W for the PC, and at 1024×768 the gap widens further, with the board achieving 0.066 fps/W against 0.027 fps/W for the PC, corresponding to an efficiency advantage of roughly 2.4 times at the lower resolution. This result follows directly from the board's power consumption being nearly two orders of magnitude lower than the PC's (1.891 W versus 143 W), a difference large enough to outweigh the board's lower absolute frame rate. The efficiency advantage is also resolution-dependent: because the number of coding units per frame, and therefore the board's per-frame latency, scales with frame area, the board's relative efficiency improves as resolution decreases, whereas the PC's efficiency scales more moderately since its processing pipeline is not similarly bottlenecked by sequential per-coding-unit evaluation. These results suggest that, although the board remains far from real-time throughput in absolute terms, its very low power envelope already makes it a favourable choice for energy-constrained deployments where frame-rate requirements are modest, and that closing the throughput gap through architectural optimisations such as coding-unit-level parallelism or deeper pipelining would further widen this efficiency advantage in the board's favour.

5) Carbon footprint: Beyond raw energetic efficiency, the substantial power gap between the two platforms also translates into a markedly different carbon footprint per unit of operating time. Using the International Energy Agency's (IEA) reported global average electricity grid carbon intensity of approximately 445 g CO_2/kWh in 2024 [21], the FPGA board's total on-chip power consumption of 1.891 W corresponds to an emission rate of roughly 0.000234 g CO_2 per second of operation, compared to approximately 0.0177 g CO_2 per second for the PC at 143 W, a difference of about two orders of magnitude consistent with the power ratio between the two platforms. Although the PC's higher throughput partially offsets this gap on a per-frame basis, the board still emits noticeably less CO_2 per unit of processing time at both resolutions, reinforcing its suitability for energy- and emissions-constrained deployment contexts, particularly as future architectural optimisations narrow the throughput gap without altering its favourable power profile.

6) Comparison with reference design: Table XI compares the key performance figures reported by Bouaafia et al. [16] for their PYNQ-Z1-based CNN_1 accelerator against those obtained for our proposed design on the PYNQ-Z2 platform, both targeting the 64×64-level CU partition prediction for HEVC.

TABLE XI. COMPARISON WITH THE REFERENCE PYNQ-Z1 CNN ACCELERATOR [16]

Metric	Bouaafia et al. [16]	Our Design
Platform	PYNQ-Z1 (XC7Z020)	PYNQ-Z2 (XC7Z020)
PL frequency (MHz)	142	54.1
LUT utilisation (%)	47	65
FF utilisation (%)	26	21
BRAM utilisation (%)	16	33
DSP utilisation (%)	14	10
On-chip power (W)	1.69	1.891
Processing time / CU	400 ms	31.3–51.8 ms

Compared with the reference design, our accelerator operates at a substantially lower clock frequency (54.1 MHz versus 142 MHz), yet achieves a processing time per 64×64 coding unit that is roughly 8 to 13 times faster (31.3–51.8 ms versus 400 ms). This indicates that, cycle efficiency aside, the reference design's higher operating frequency does not translate into a comparable advantage in absolute latency, most likely due to differences in interface overhead and inter-IP data movement between the streaming architecture and the PS in [16]. In terms of resource utilisation, our design consumes more LUTs (65% versus 47%) and more BRAM (33% versus 16%), reflecting the additional control and multiplexing logic associated with the higher degree of arithmetic resource sharing discussed earlier, while using fewer FFs (21% versus 26%) and notably fewer DSPs (10% versus 14%). On-chip power consumption is comparable between the two designs (1.891 W versus 1.69 W), with our design drawing marginally more power, which is consistent with its higher LUT occupancy. Overall, these results suggest that our design trades a modest increase in logic resource usage and power for a considerably lower per-CU processing latency, positioning it favourably relative to the reference work in terms of raw throughput, even though both designs remain well below the requirements of real-time HEVC encoding at typical video resolutions.

D. Discussion

When read together, the results above reflect the same trade-off at two levels. In software, the heuristic and machine-learning methods reviewed in Section II [11], [13] cut RDO calls through fixed or learned decision rules, in hardware, the accelerator cuts clock cycles and memory traffic through the identical rule, so the complexity reduction in Table VI and the latency spread in Table VIII are two views of the same early-termination mechanism. Relative to [16]'s multi-IP design, consolidating the network into a single IP core is what drives the latency advantage rather than clock frequency, which is lower here (54.1 versus 142 MHz), the LUT/DSP imbalance this incurs (65% versus 10%, Table IX) is the resource cost of sharing a handful of floating-point units across the network's weights, and is consistent with the low-power, sub-real-time profile in Table X. Overall, the accelerator is best read as evidence that early-termination CNN inference can be mapped onto genuine conditional hardware execution at a favourable power budget, with throughput the main gap left to close. We note that a broader comparison against other FPGA-based CNN accelerators beyond [16] is limited by the availability of full, citable results for those designs, Section II can be extended with additional hardware baselines when new works become available.

VII. CONCLUSION

In this article, a hardware-software architecture based on a deep convolutional neural network has been proposed for the prediction of coding units in 3D-HEVC on FPGA–SoC. The presented work aims to accelerate convolutional neural networks due to their high computational load. Nevertheless, a hardware IP core for MD-CNN was developed using the Vivado HLS tool. Subsequently, an architecture integrating hardware and software was developed by incorporating hardware IP cores based on the PYNQ-Z2 board. The simulation results revealed that our design achieves an on-chip power

consumption of 1.891 W. Our design requires reduced consumption of hardware resources: 65% of LUTs, 21% of FFs, 33% of BRAMs, and 10% of DSPs. As future work, we plan incorporate additional encoder modules in the hardware implementation, in particular the intra-prediction and transform modules.

REFERENCES

[1] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3d extensions of high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.

[2] G. Sanchez, L. Agostini, and C. Marcon, "Algorithms for efficient and fast 3d-hevc depth map encoding," 2020.

[3] N. Omran, A. Maraoui, I. Werda, and B. Hamdi, "Fast partition algorithm in depth map intra coding unit based on multi-deep convolution neural network," *Journal of Real-Time Image Processing*, vol. 21, no. 1, p. 23, 2024.

[4] H. Hamout and A. Elyousfi, "Fast 3d-hevc pu size decision algorithm for depth map intra-video coding," *Journal of Real-Time Image Processing*, vol. 17, no. 5, pp. 1285–1299, 2020.

[5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] S. Bakkouri and A. Elyousfi, "An adaptive cu size decision algorithm based on gradient boosting machines for 3d-hevc inter-coding," *Multimedia Tools and Applications*, pp. 1–19, 2023.

[7] Z. Zhang, L. Yu, J. Qian, and H. Wang, "Learning-based fast depth inter coding for 3d-hevc via xgboost," in *2022 Data Compression Conference (DCC)*. IEEE, 2022, pp. 43–52.

[8] W. Song, P. Dai, and Q. Zhang, "Content-adaptive mode decision for low complexity 3d-hevc," *Multimedia Tools and Applications*, pp. 1–16, 2023.

[9] J. Y. Lee, M. Kang, and S.-h. Park, "Fast depth intra mode decision using machine learning in 3d-hevc," *Available at SSRN 4197680*.

[10] J.-C. Chiang, K.-K. Peng, C.-C. Wu, C.-Y. Deng, and W.-N. Lie, "Fast intra mode decision and fast cu size decision for depth video coding in 3d-hevc," *Signal Processing: Image Communication*, vol. 71, pp. 13–23, 2019.

[11] H. Hamout and A. Elyousfi, "A computation complexity reduction of the size decision algorithm in 3d-hevc depth map intracoding," *Advances in Multimedia*, vol. 2022, p. 3507201, 2022.

[12] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Fast 3d-hevc depth map encoding using machine learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 850–861, 2019.

[13] C.-H. Fu, H. Chen, Y.-L. Chan, S.-H. Tsang, H. Hong, and X. Zhu, "Fast depth intra coding based on decision tree in 3d-hevc," *IEEE Access*, vol. 7, pp. 173 138–173 147, 2019.

[14] C. Liu, K. Jia, and P. Liu, "Fast depth intra coding based on depth edge classification network in 3d-hevc," *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 97–109, 2021.

[15] H. Dehnavi, M. Dehnavi, and S. Haghzad Klidbary, "Fcd-cnn: Fpga-based cu depth decision for hevc intra encoder using cnn," *Journal of Real-Time Image Processing*, vol. 21, p. 105, 2024.

[16] S. Bouaafia, R. Khemiri, S. Messaoud, and F. E. Sayadi, "Deep cnn co-design for hevc cu partition prediction on fpga–soc," *Neural Processing Letters*, vol. 54, no. 4, pp. 3283–3301, 2022.

[17] ITU/ISO/IEC, "3D-HEVC HTM reference software, version HTM-16.3," 2017, accessed: May 25, 2026. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/branches/HTM-16.3-fixes/

[18] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM multimedia systems conference*, 2015, pp. 219–224.

[19] ITU/ISO/IEC, "HEVC HM reference software, version HM-16.18," 2017, accessed: May 25, 2026. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tree/HM-16.18

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[21] International Energy Agency, "Electricity mid-year update 2025," 2025, accessed 2026. [Online]. Available: https://www.iea.org/reports/electricity-mid-year-update-2025/emissions-power-generation-co2-emissions-are-plateauing

The Coordination Complexity Index: A Parameter-Free Checkpoint Diagnostic for Silent Coordination Failure in Cooperative Multi-Agent Reinforcement Learning

Chatchitsanu Pothisakha[1], Pudsadee Boonrawd[2]*, Siranee Nuchitprasitchai[3]
Data Science for Innovation Program-Faculty of Information Technology and Digital Innovation,
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand[1]
Department of Information Technology-Faculty of Information Technology and Digital Innovation,
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand[1,2,3]

***Abstract*—Cooperative multi-agent reinforcement learning (MARL) can fail silently while standard reward and loss curves remain inconclusive. This study introduces the Coordination Complexity Index (CCI), a checkpoint statistic with no learned parameters that summarizes dispersion among per-agent greedy Q-values. A frozen checkpoint cannot by itself establish a temporal failure; it can reveal Q-value patterns that motivate further inspection. On Level-Based Foraging (LBF), rules calibrated on the VDN/QMIX/QTRAN triad assign operational high-occupancy, dual-occupancy, and near-zero-dispersion labels, respectively. These labels are not independent proof of stagnation, oscillation, or behavioral roles. The pooled collapsed cohort differs from the non-collapsed LBF cohort by more than four orders of magnitude at the cohort geometric mean, while QMIX attains higher return with lower coordination variance than VDN. Held-out QPLEX falsifies the proposed class-complete-IGM prediction: return is zero, yet C does not approach the collapse cut-off. At 10% of the canonical LBF budget, a retrospective $\bar{C} < 10^{-3}$ rule flags all retained QTRAN seeds without flagging retained QMIX seeds; on RWARE it produces at least one environment-induced false positive. CCI is therefore a candidate domain-specific checkpoint screen, not a universal early-stop rule or a coordination-quality score. With the cut-off frozen before evaluation, 5/5 additional QTRAN training seeds remain below it at the retained 10%-budget checkpoint and 5/5 remain below it near 10M steps; this is a positive-class repeatability check, not independent calibration or a new specificity control.**

Keywords*—Cooperative Multi-Agent Reinforcement Learning (MARL); value decomposition; coordination signatures; coordination collapse; Coordination Complexity Index (CCI); checkpoint screening; Centralized Training with Decentralized Execution (CTDE)

I. Introduction

A cooperative multi-agent reinforcement learning (MARL) experiment can consume days of training compute yet fail silently: even without outright collapse, the loss may keep falling and team return may keep rising after the joint policy has already settled into a degenerate, task-irrelevant regime. Because a cooperative team maximizes a shared return under partial observability and decentralized execution, neither the reward curve nor any single agent's local view necessarily exposes that failure.

Value-decomposition (VD) methods, a leading family for cooperative training, combine per-agent Q-values through centralized mixers with progressively relaxed constraints, from VDN and QMIX to QTRAN and QPLEX [1]–[4]. This progression suggests an implicit assumption: greater mixer expressiveness should yield at least equal team performance. That assumption is tested directly with a concrete counterexample. On the canonical Level-Based Foraging (LBF) 9×9-5p grid, QTRAN collapses to zero return (0.000), whereas QMIX reaches 0.732. Thus, greater expressiveness alone does not prevent coordination failure. In single-agent deep learning, a network can quietly lose the ability to learn even as gradient-descent training continues [5]. Cooperative MARL similarly lacks a low-cost checkpoint observable for inspecting silent degeneration. Table I positions the proposed scalar against related diagnostics, but the comparison is qualitative rather than an empirical head-to-head. Reward curves can obscure whether disagreement is useful, and return is not a substitute for the proposed coordination proxy C: QMIX earns a higher return than VDN with lower coordination variance, consistent with the externally reported sample-efficiency gain under monotonicity [6].

This study makes three contributions.

- A checkpoint diagnostic with no learned parameters. C, the Coordination Complexity Index [CCI, Eq. (2)], is a per-timestep scalar computed from the variance of per-agent Q-values without an auxiliary loss. Its classifier uses thresholds calibrated on the LBF VDN/QMIX/QTRAN triad. The diagnostic can be extracted from a frozen checkpoint and is provided as the `macci` package described in the Data and Code Availability statement.
- A three-pattern operational taxonomy. On LBF, VDN, QMIX, and QTRAN exhibit high occupancy, dual occupancy, and near-zero dispersion under the calibrated rules. Only the QTRAN pattern transfers unchanged to RWARE. The conventional labels Stagnation,

*Corresponding author.

BI-REGIME, and COLLAPSE are retained for reproducibility, but they describe checkpoint statistics rather than independently verified temporal or behavioral mechanisms. On seed-matched cohorts from QFix, QTypeMix, and DuelMIX, C also separates successful dual-occupancy seeds from failed one-band seeds; this is partial rather than universal cross-architecture evidence.

- A held-out evaluation that bounds where C applies. The held-out prediction is that a *class-complete* Individual-Global-Max (IGM) architecture reaches the $\bar{C} < 10^{-3}$ collapse cut-off by the fixed 10M-step checkpoint. On five held-out QPLEX seeds that prediction fails: $\bar{C}$ remains heterogeneous rather than collapsing. The accompanying return collapse ($5/5$ seeds at 0.000) is reported separately. The result indicates that C tracks coordination-variance collapse rather than every form of policy failure.

II. RELATED WORK

Value-decomposition methods organize cooperative training around a centralized mixing function that recombines per-agent Q-values, and the family is usually read as a ladder of increasing expressiveness. VDN sums the per-agent values, satisfying the Individual-Global-Max (IGM) condition trivially but representing no inter-agent interaction [1]. QMIX replaces the sum with a state-conditioned monotonic mixing network whose non-negative weights preserve IGM while admitting nonlinear interactions [2]. QTRAN and QPLEX climb further, the former relaxing monotonicity through auxiliary losses and the latter through a duplex dueling advantage formulation, so as to cover a strictly larger IGM function class [3], [4]. The organizing premise of this ladder is that broader function-class coverage yields at-least-equal team performance.

The 2024–2026 literature continues to climb the same ladder rather than questioning the premise. DuelMIX, QTypeMix, and QFix extend factorization for heterogeneous teams or incomplete base mixers [7]–[9]; related fixed mixers reshape the mixing algebra itself (including QVF, which integrates quantile regression into the IGM condition) [10]–[15]. Each mixer above is fixed at design time and unchanged during training, and none poses the complementary question of what to measure when added expressiveness does not translate into performance—the question taken up here.

A second thread supplies the contrary empirical evidence that motivates a diagnostic. Rashid et al. established monotonic mixing as the practical default on cooperative StarCraft (SMAC) [2], and Hu et al. later showed that a carefully tuned QMIX matches the SMAC state of the art, with the monotonicity constraint improving sample efficiency on purely cooperative tasks [6]. This inverts the ladder's premise, since the constraint that the more expressive methods discard may itself be doing useful work. Consistent with that reading, value decomposition provably cannot represent the global value function on symmetric multi-modal reward landscapes, for every IGM-consistent mixer and whether or not the local networks are shared [16]. Linear and monotonic factorizations, in turn, cannot guarantee consistency between per-agent greedy actions and the team-optimal joint value [17]. Empirically, Papadopoulos et al. report that QPLEX fails with near-zero return on the hardest LBF tasks [18]—a silent collapse that registers in neither the training loss nor a cursory return summary until the compute has already been spent. Standardized cooperative-evaluation protocols [19], [20] improve reporting but still do not contribute a single observable that diagnoses coordination structure from a checkpoint.

The third thread concerns direct measurement of inter-agent coordination, which remains scarce relative to mixer design. Parameter-free probes such as Role Diversity [21], System Neural Diversity [22], CASEC [23], the Specialization Index [24], LVMQ [25], entropy and update probes [26], neuron statistics [27]–[29], greedy-action churn [30], and action-value gaps [31] characterize related but different objects. This study does not implement the published Role Diversity, policy-churn, or neural-diversity diagnostics under a shared protocol; Table I records their stated scope and instrumentation requirements.

Closest in spirit is Yamaguchi [32], who identifies coordinated-stable, fragile, and jammed/disordered regimes using cooperative success rate and TD-error variance. That method uses two training-time axes, whereas CCI reads Q-value dispersion from a frozen checkpoint. The present study targets value decomposition and evaluates LBF and RWARE, but the observed cross-domain split is associative rather than causal. Concurrent work reports that similar return curves can conceal distinct coordination mechanisms [33]. Table I is a qualitative positioning table, not evidence that CCI outperforms Role Diversity, phase-regime analysis, entropy, churn, or Q-gap diagnostics. Joint TD-error and final return are the matched main-protocol alternatives; a post-hoc retained-checkpoint comparison with entropy, Q-gap, and two behavioral summaries is reported separately in Section V-A. Cross-domain support is marked Partial because only the QTRAN near-zero-dispersion pattern transfers unchanged to RWARE. The all-Yes entries for CCI are therefore scoped to the properties listed and do not constitute an exhaustive priority claim.

TABLE I. C (CCI) VS. PRIOR COORDINATION DIAGNOSTICS

Diagnostic	Param-free	No instr.	VD-collapse	Cross-domain	Held-out
Role Diversity [21]	Yes	Yes	No	Yes	No
Phase regimes [32]	Yes	No	No	No	No
Joint TD-error (Q_{tot}) [34]	Yes	No	No	–	No
Final return [19]	Yes	Yes	No	–	No
C (CCI, this work)	**Yes**	**Yes**	**Yes**	**Partial**	**Yes**

III. METHODOLOGY

The coordination diagnostic C has no learned parameters and is extractable from a frozen checkpoint. It is designed to discriminate the VDN/QMIX/QTRAN calibration triad on $\bar{C}$, with a return-based check retained for class-complete IGM mixers such as QPLEX. Its 10%-budget use is evaluated retrospectively on retained LBF QMIX/QTRAN checkpoints and is treated as a candidate screen rather than a validated stopping policy.

A. Setting and the Value-Decomposition Family

Cooperative MARL is cast as a Dec-POMDP under Centralized Training with Decentralized Execution (CTDE) [1],

[2], [35]: joint value decomposition is trained with global state and histories available, while each agent executes from its own history τ_i via $Q_i(\tau_i, u_i)$. An action-value decomposition satisfies the Individual-Global-Max (IGM) condition [2], [3] when:

$$\arg\max_{\mathbf{u}} Q_{\text{tot}}(\boldsymbol{\tau}, \mathbf{u}, s) \;=\; \big(\arg\max_{u_i} Q_i(\tau_i, u_i)\big)_{i=1}^{n}. \quad (1)$$

Eq. (1) licenses decentralized greedy execution. The value-decomposition family realizes it with decreasing mixing restrictiveness: VDN [1] sums per-agent utilities; QMIX [2] uses a state-conditioned mixer with non-negative weights (monotonicity $\partial Q_{\text{tot}}/\partial Q_i \geq 0$); QTRAN [3] drops monotonicity and approximates the full IGM-consistent class via auxiliary losses; QPLEX [4] attains the *complete* IGM function class with a duplex-dueling architecture. This VDN→QMIX→QTRAN→QPLEX axis is the structural ladder along which coordination behavior is measured below.

B. CCI Formula and Components

Given the IGM-consistent value-decomposition family above, a frozen checkpoint permits inspection of whether per-agent greedy Q-values remain numerically heterogeneous or become nearly identical. CCI reads that numerical property through the variance of per-agent greedy-action Q-values, passed through a tanh wrapper. It does not by itself identify behavioral roles, coordination success, or a causal failure mechanism. The **Coordination Complexity Index (CCI)**, denoted C, is:

$$C \;=\; \tanh\big(\mathrm{Var}\big(q_1, \ldots, q_n\big)\big) \quad (2)$$

where, $\mathbf{q} = (q_1, \ldots, q_n)$ is the vector of per-agent greedy-action Q-values at a given timestep and Var is the unbiased sample variance over the n agents (dividing by $n-1$; ddof=1). That is the estimator which produced every C value reported in this study, and reproducing the reported numbers requires it. The `macci` implementation uses the same estimator, so the package and the reported values agree by construction. The population variance (ddof=0) differs only by the constant factor $n/(n-1)$ on the pre-tanh argument, but substituting it under the same fixed thresholds is not label-neutral: two of the twenty-five LBF seeds cross the STAGNATION boundary (VDN $23/25 \to 21/25$), while the cohort-level signature of each algorithm is unchanged.

High C indicates dispersed per-agent utilities; it does not by itself imply role differentiation, adaptation, or success. Likewise, $C \approx 0$ denotes nearly identical greedy utilities under the realized scale, not every possible form of policy collapse. For illustration, $\mathbf{q} = (2.0, 2.0, 2.0)$ gives $C = 0$, whereas $\mathbf{q} = (0.5, 2.0, 3.5)$ gives $C = \tanh(1.5) \approx 0.91$. A common additive shift $q_i' = q_i + b$ leaves C unchanged, but a common positive rescaling gives $C' = \tanh(a^2 \,\mathrm{Var}(\mathbf{q}))$ and can change every finite threshold decision. Agent-specific action-independent offsets can also preserve each agent's greedy policy while changing inter-agent variance. Thus the tanh wrapper bounds $C \in [0, 1)$ but does not make it policy- or scale-invariant.

The headline scalar C above is one component of a richer per-timestep (I, C, T) triple logged by the artifact. Here $I = \sigma(\mathrm{mean}_{i \neq j} |(q_i - \bar{q})(q_j - \bar{q})|)$ is an interdependence proxy computed from pairwise products of centered selected-action values; it is not softmax entropy. C is the Eq. (2) scalar, and T is an L2 trajectory-change term. T is zero in the post-hoc single-checkpoint extraction because no preceding trajectory is supplied. I and C are non-orthogonal functionals of the same selected-action values, so the component analysis below is descriptive rather than a comparison of independent diagnostics.

C. Signature Extraction Pipeline

The signature-extraction routine records C at every timestep of greedy evaluation episodes ($\epsilon = 0$). A timestep is LOW if $C < \theta_{\text{lo}}$ and HIGH if $C > \theta_{\text{hi}}$; the neutral band is unmarked. A regime transition is counted when consecutive timesteps lie on opposite sides of θ_{lo}, and the reported rate is the episode mean. The operational classifier is applied in order: COLLAPSE if $\bar{C} < 10^{-3}$; STAGNATION if $\max(\text{High}\%, \text{Low}\%) > 80\%$; and BI-REGIME otherwise. These names preserve the registered analysis vocabulary. Occupancy ignores temporal order, so STAGNATION means one-band occupancy and BI-REGIME means dual-band occupancy; neither label alone establishes halted adaptation or persistent oscillation. Transitions are reported separately and do not enter classification. The constants were calibrated on the LBF VDN/QMIX/QTRAN triad and were not re-tuned per environment. "No learned parameters" refers only to C, which has no fitted weights or auxiliary loss.

1) Threshold robustness: The cohort label is unchanged across five $(\theta_{\text{lo}}, \theta_{\text{hi}})$ settings: $(0.03, 0.07)$, $(0.04, 0.08)$, $(0.05, 0.10)$, $(0.08, 0.15)$, and $(0.10, 0.20)$. QTRAN is COLLAPSE in all $25/25$ seed-by-threshold cells because occupancy thresholds do not enter that decision. A sensitivity check of collapse cut-offs $\{10^{-4}, 5{\times}10^{-4}, 10^{-3}, 2{\times}10^{-3}, 10^{-2}\}$ also leaves every canonical QTRAN seed below and every VDN/QMIX seed above. This analysis varies numerical cut-offs on the same calibration triad; it does not validate the rule form on an independent calibration set. VDN is STAGNATION in $23/25$ cells and QMIX is BI-REGIME in $17/25$, so the non-collapse boundary is cohort-level rather than seed-universal. The retained sweep enables the same sensitivity check on new checkpoints. A separate post-hoc rescaling audit applies $q' = aq$ exactly through the retained pre-tanh variances. At $a = 0.1$, three of five QMIX seeds cross the collapse cut-off even though greedy actions are unchanged; at $a = 10$, all five QMIX seeds become one-band high-occupancy cases. QTRAN remains below the collapse cut-off for all five seeds over $a \in [0.1, 10]$ because its observed variance is much smaller. This finite-range robustness of the reported QTRAN cohort does not restore policy invariance to the statistic or its thresholds.

D. Dual Evaluation Protocol

The tanh normalization in Eq. (2) assumes $|Q| \lesssim 1$, which holds for environments with bounded episodic return and short horizons. On reward-unbounded environments where $|Q|$ can exceed unity by an order of magnitude or more, C saturates to one regardless of inter-agent coordination structure. Two layers

of diagnostic are therefore defined. Layer 1 is the canonical C of Eq. (2). Layer 2 is its un-bounded counterpart:

$$\mathrm{RawVar}(Q) = \mathrm{Var}_i\big(\max_{u_i} Q_i(\tau_i, u_i)\big), \tag{3}$$

and is reported together with the per-state $\max_i Q_i - \min_i Q_i$ range; applying the tanh wrapper to Eq. (3) recovers Layer 1, so the two layers differ only in whether the dispersion is compressed. Each layer targets the same numerical property—dispersion among per-agent selected-action values—in the regime where its observable remains informative. All headline tables use Layer 1 on the bounded-reward LBF and RWARE cohorts; Layer 2 is defined for reward-unbounded regimes where tanh saturates, and is not reported as an empirical probe in this study. Signature extraction runs a fresh-evaluation protocol on the frozen checkpoint under evaluation: 100 greedy episodes ($\epsilon = 0$), reduced to 50 for some early-checkpoint extractions (Table VII), with evaluation seeds independent of the training seeds. It therefore depends on no quantity logged during training.

E. Algorithms and Experimental Setup

This work studies three primary value-decomposition algorithms: VDN, QMIX and QTRAN. QPLEX, held out from the design of C, is *class-complete*: it realizes the complete IGM function class as a hard architectural constraint rather than through auxiliary penalties. The three primary algorithms train from one shared hyperparameter configuration on the LBF grids at a learning rate of 5×10^{-4}. Two things are not shared: the optimiser is Adam for VDN and QMIX and RMSprop for QTRAN [36], [37], and the QTRAN learner does not consume every key of that configuration (it ignores `td_lambda`, the leniency keys, `q_value_clip` and `standardise_rewards`). The mixer is therefore the primary intended per-algorithm difference, not the only realized one. The RWARE cohorts run to 2×10^7 steps. Every setting is reported in full, as standardised evaluation requires [19]; the shared table is Table V (batch size / replay-buffer 128 / 5000 episodes; agent network Feed-forward MLP, hidden dim 64). One setting is *not* shared: the optimiser. Rather than leave that optimiser asymmetry as a confound, a matched QTRAN-with-Adam control (n=5 seeds) was run to separate the mixer from the optimiser; without it, QTRAN would have been both the sole COLLAPSE class and the sole RMSprop run, and the two would not have been separable. The shared hyperparameters follow Hu et al.'s implementation discipline [6], using code-level tricks that are not specific to QMIX. As executed they are not generic across the family either: the QTRAN learner ignores them.

1) Held-out prediction H1: Prediction H1 is an exploratory, falsifiable conjecture tied to the training budget. Let $\mathcal{A}$ realize the complete IGM function class through a hard architectural constraint. Under assumptions (A1)–(A3), H1 predicts that every seed reaches $\bar{C} < 10^{-3}$ by the fixed 10M-step checkpoint on canonical LBF. The assumptions are not identified from the reported data or derived in general, so H1 is neither a theorem nor a convergence claim. VDN, QMIX, and QTRAN supplied the taxonomy and thresholds; QPLEX is only a held-out falsification test of that calibration.

The working assumptions are:

- (A1) parameter-shared agent networks with a symmetric team reward $r(s, \mathbf{u})$;
- (A2) the TD fixed point under parameter-shared training admits the additive-plus-shared decomposition $\max_{u_i} Q_i(\tau_i, u_i) = f(\tau_i) + c(s)$ for a shared function f and a state-level residual $c(s)$ (a structural assumption, not derived here); and
- (A3) a gradient-descent training-limit regime under which parameter-shared updates pull the shared function f of (A2) toward a common attractor on the visited history distribution (a regularity assumption, not derived here, in the spirit of the gradient-flow analysis of T. Hu et al. [38]). The agent networks receive a one-hot agent identifier (Table V), which breaks exact exchangeability of the local histories by construction; (A3) is therefore a statement about the learned representation in the training limit, not about the raw histories. In the configuration studied here the agent network is feed-forward (`use_rnn=False`; Table V), so the realized per-agent function is $Q_i(z_i, u_i)$ over the current observation; τ_i is retained for consistency with the CTDE formalism.

The motivation is heuristic. Under (A1), shared TD targets [2], [3] and (A2) yield $\max_{u_i} Q_i = f(\tau_i) + c(s)$, hence $\mathrm{Var}_i(\max_{u_i} Q_i) = \mathrm{Var}_i(f(\tau_i))$. Only if representation sharing under (A3) dominates should this variance contract. These assumptions are neither identified empirically nor sufficient for a general proof. Consequently, H1 is evaluated solely as a budget-anchored conjecture: one QPLEX seed with $\bar{C} \geq 10^{-3}$ at 10M steps rejects it.

2) Environments and seed counts: The primary benchmark is Level-Based Foraging (LBF) [34], [39]. Five primary 5-agent grids (8×8 through 12×12) plus a 6-agent 9×9 robustness probe are tested, all with three food items, `force_coop=True`, and sight radius two. Those settings are denoted by the compact labels 9×9-5p-3f-coop and 9×9-6p; the 9×9-5p grid on which C, its thresholds, and the held-out test are defined is hereafter the canonical grid. The seed count of every reported cohort is stated explicitly, and the per-seed records are available from the corresponding author on reasonable request. On LBF, VDN/QMIX/QTRAN each contribute 5 matched seeds per grid, so the headline taxonomy rests on the 25 QTRAN seeds across the five primary grids. VDN and QMIX were run with more seeds, but the shared 5 form the matched reported cohort, consistent with the 5-vs-5 effect sizes of Table III. The 6-agent 9×9-6p variant likewise reports 5 seeds per algorithm, and the held-out QPLEX test uses 5 seeds on the canonical grid. Cross-domain validation on RWARE [40] (the four-agent `rware-small-4ag` layout) uses 5 seeds per algorithm and 20M environment steps under the parallel runner. The cohort is matched on seed count and budget but not on the leniency schedule or batch size (32 for VDN and QTRAN against 128 for QMIX). A second matched n=5 cohort is retrained under the episode runner, which collects one full episode at a time. Reported returns are undiscounted, environment-specific episodic metrics [40]: on LBF the fraction of food items collected per episode, on RWARE the number of successful deliveries per episode. LBF

TABLE II. THREE COORDINATION SIGNATURES ON LBF 9×9-5P

Alg.	Signature	$\bar{C}$	Tr/ep	Lo%	Hi%	Return	95% CI
VDN	Stagnation	0.483	1.2	3.9	91.8	0.632	[0.55, 0.71]
QMIX	Bi-regime	0.172	2.4	22.8	57.2	0.732	[0.70, 0.76]
QTRAN	Collapse	3.3×10^{-6}	0.0	100.0	0.0	0.000	[0.000, 0.000]

and RWARE return magnitudes are therefore not on a common scale.

3) Statistical protocol: Effect sizes use Cohen's d [41] with the Hedges' g small-sample correction [42]

$$J(df) = 1 - \frac{3}{4(n_1 + n_2) - 9}. \quad (4)$$

Confidence intervals are percentile bootstrap intervals over the per-seed values, computed directly in NumPy with a fixed seed: 10,000 resamples for effect-size intervals and 50,000 for return intervals and $P(X > Y)$; the summaries reported—the interquartile mean (IQM), optimality gap, and $P(X > Y)$—follow the reporting practice argued for by Agarwal et al. [43], but the stratified-bootstrap procedure of the `rliable` library itself was not used. Where $n \leq 3$ makes $J(df)$ unreliable, the row is footnoted. Wide bootstrap CIs from near-zero within-group variance (collapsed conditions) are reported unadjusted. All seed-level statistics (CIs, effect sizes) are computed over the per-seed $\bar{C}$ values—each seed's $\bar{C}$ being the mean of C over all timesteps pooled across its evaluation episodes—so the reported spread reflects across-seed reproducibility.

IV. RESULTS

Results cover the canonical signatures, a frozen-cut-off extension on five additional QTRAN training seeds, cross-setting robustness, an optimiser control, generalization to unseen architectures, and a held-out evaluation on QPLEX.

A. Three Coordination Signatures

The CCI extraction routine uses 100 greedy episodes per seed on canonical LBF. VDN receives the operational STAGNATION label because HIGH occupancy is approximately 92%; this indicates a stable one-band Q-value profile, not verified behavioral stagnation or fixed roles. QMIX receives BI-REGIME because trajectories occupy both bands, with approximately 2.4 threshold crossings per episode; the statistic does not establish what agents do on either side of the threshold. QTRAN receives COLLAPSE because greedy Q-values are nearly identical and LOW occupancy is 100%. Its seed-mean $\bar{C}$ is nearly five orders of magnitude below QMIX (Table II). Returns and bootstrap intervals use the matched five-seed cohort. Return and C are aligned by training seed but come from training logs and separate greedy checkpoint evaluations, respectively; they are not paired episode-level observations. Canonical QMIX uses last-saved rather than strictly end-of-budget checkpoints, as disclosed in Section V-C.

These operational labels are cohort-level summaries rather than artifacts of one threshold setting. Per-seed stability is exact for COLLAPSE and majority-stable for the other labels: VDN is STAGNATION in 23/25 cells and QMIX is BI-REGIME in 17/25 across the five threshold pairs. Attribution to the mixer rather than the optimiser rests on Section IV-D. Table X adds a descriptive seed-0 cross-checkpoint trace. VDN transitions fall from 6.60 at 2.5M steps to 0.14 at 5.0M and 7.5M; QMIX remains between 2.38 and 4.66 across six checkpoints; QTRAN remains at zero across six checkpoints. This limited trace preserves temporal ordering for one retained seed but is not a cohort-level temporal validation.

Neither final return nor joint TD-error identifies the observed Q-value-dispersion pattern as directly as C. At the 10% canonical-LBF checkpoint, the retrospective candidate rule $\bar{C} < 10^{-3}$ flags all five retained QTRAN seeds and none of the five retained QMIX seeds. VDN checkpoints were not retained there, so the evidence covers one grid and two algorithms. The flagged LBF seeds did not recover by the retained final checkpoint. On RWARE, however, two of three retained early QMIX checkpoints are flagged, including at least one seed that later succeeds. The same cut-off therefore has an environment-induced false positive and is not a validated universal stop rule.

1) Frozen-cut-off seed extension: After fixing the rule and evaluation protocol, five additional canonical-LBF QTRAN training seeds (5–9) were run under the same 10M-step parallel protocol. Under the same 100-episode greedy harness, 5/5 seeds fall below 10^{-3} at the nearest retained 10%-budget checkpoint (cohort mean $\bar{C} = 3.93\times10^{-6}$), and 5/5 do so near 10M steps (mean $\bar{C} = 5.27\times10^{-6}$). Fresh-evaluation mean returns are 0.001 and 0.006, respectively. Table XII reports every seed. Because this extension adds only QTRAN, it tests within-architecture repeatability of the positive class; it neither recalibrates the cut-off nor supplies new negative-class specificity evidence.

B. Statistical Separation

The three architecture cohorts on the canonical grid separate statistically on seed-level $\bar{C}$. Table III reports pairwise effect sizes on seed-level $\bar{C}$ (Cohen's d with the Hedges' g correction of Eq. (4); bootstrap CIs on d from the same unstratified percentile bootstrap, 10,000 resamples with a fixed seed). VDN-versus-QTRAN separates at $|d| = 13.46$ ($g = 12.15$), a magnitude that reflects QTRAN's near-zero within-group variance under collapse. Two of the three pairs exceed $|d| = 2.0$, with QMIX-versus-QTRAN ($d = 1.73$) just below. This holds at the point estimate; at $n = 5$ the QMIX-versus-VDN interval extends to $|d| = 1.74$, so the $|d| = 2.0$ boundary is unresolved for that pair, as it is for QMIX-versus-QTRAN, whose interval runs from 1.36 to 10.18. All three intervals exclude zero, so the direction of every pair is resolved, but each spans a factor of six or more in $|d|$, so at $n = 5$ this sample fixes the sign of each separation without fixing its size. The widest CI, $[9.69, 61.44]$ on VDN-versus-QTRAN, reflects QTRAN's near-zero within-group variance under bootstrap rather than point-estimate instability, and is reported transparently rather than narrowed via a non-standard estimator. On the same matched $n = 5$ cohort, final return separates QMIX from VDN at $d = +1.60$ and joint TD-error at $d = -1.31$; Discussion interprets these margins against C.

These pairwise effects generalize to a population-level summary. Comparing non-collapsed $\bar{C}$ on the canonical 9×9-5p cohort (ten VDN/QMIX seed-mean values; geometric mean

≈ 0.252) against collapsed $\bar{C}$ pooled across QTRAN on the canonical 9×9-5p grid and QTRAN on cross-domain RWARE small-4ag (geometric mean $\approx 5.59\times10^{-6}$) defines the pooled contrast. The geometric-mean separation is approximately 4.5×10^{4}, or about 4.66 orders of magnitude relative to that non-collapsed foraging geometric mean. Only QTRAN's RWARE values enter this collapsed pool; the VDN/QMIX RWARE cohort stays non-collapsed (below). The worst-case two-point extremes ratio—the smallest non-collapsed $\bar{C}$ (0.0613, QMIX seed 3) over the largest collapsed one (1.92×10^{-4}, RWARE QTRAN seed 2)—is 3.2×10^{2}, about 2.50 orders of magnitude. It is a two-point statistic; leaving out either extreme moves it only to 4.0×10^{2} or 4.3×10^{2}. The retained per-seed checkpoint records used here are available from the corresponding author on reasonable request.

TABLE III. PAIRWISE EFFECT SIZES ON SEED-LEVEL $\bar{C}$ (LBF 9×9-5P)

Pair	**Cohen's** d	**Hedges'** g	95% **CI on** d	n **seeds**
QMIX vs QTRAN	1.73	1.56	$[1.36, 10.18]$	5 vs 5
VDN vs QTRAN	13.46	12.15	$[9.69, 61.44]$	5 vs 5
QMIX vs VDN	−2.93	−2.65	$[-16.42, -1.74]$	5 vs 5

Across the five 5-agent grids the per-component breakdown of I, C [Eq. (2)], and T attributes 93–98% of the attributable within-cohort variance to C, versus 2–7% to I. T is structurally zero under single-checkpoint extraction—temporal drift needs a previous trajectory—so it cannot explain variance here; that is why C is reported as the single headline diagnostic. On the canonical 9×9-5p grid the VDN/QMIX/QTRAN triad alone spans 0.483 on C (comparable to the other grids); including the held-out QPLEX raises the four-architecture C spread to 0.541 versus 0.163 on I (a 3.3× C/I ratio that mixes the calibration triad with the held-out test). Because I and C are non-orthogonal functionals of the same Q-vectors, this dominant share is a descriptive corroboration of the C/I ratio rather than an orthogonal variance analysis.

C. Cross-Setting Robustness

The canonical-grid finding replicates across five LBF grid sizes and a six-agent variant (per-grid QMIX/VDN $\bar{C}$ and returns in Table VIII). VDN's mean $\bar{C}$ stays above 0.47 on all five 5-agent grids and on the 6-agent variant; QMIX's ranges from 0.083 at 8×8 to 0.208 at 10×10 across the five 5-agent grids, rising to 0.220 on the 6-agent variant; QTRAN's remains below 10^{-3} on all five 5-agent grids and on the 6-agent variant. On all six configurations, C identifies a near-zero-dispersion QTRAN pattern that return alone does not distinguish, since QTRAN's mean return ranges from exactly zero on the canonical grid to 0.029 (0.000–0.029) rather than being uniformly zero (the six-agent case is detailed below). The QMIX-versus-VDN effect size on $\bar{C}$ is non-monotonic with an intermediate-grid-size minimum: $|d_C|$ drops from 10.10 at 8×8 to 1.47 at 10×10, then rebounds to 12.78 at 12×12. The intermediate minimum is driven by a single QMIX seed at $\bar{C} = 0.660$, which raises the QMIX across-seed standard deviation at 10×10 to 0.25 against 0.05 for VDN; at the two extremes both cohorts are tight (across-seed standard deviations between 0.022 and 0.056), so the pooled standard deviation is small and $|d_C|$ is correspondingly large. The 9×9-5p effect size is $d = -2.93$ ($g = -2.65$) at $n = 5$ versus 5. On returns, the QMIX-vs-VDN gap is 9.1–16.9 pp across grids (non-monotonic, peaking at 10×10), with $P(\text{QMIX} > \text{VDN}) \geq 0.91$ on every grid (the smallest, 0.92 on 10×10, is 0.9186 before rounding). All reported cross-grid effect sizes and returns use $n = 5$ seeds per algorithm on every grid, the six-agent 9×9-6p variant included, with two exceptions: the 9×9 $P(X > Y)$ uses six QMIX seeds, and on the 6p variant the $\bar{C}$ and return columns do not share the same five seeds.

Across the five 5-agent grids, every tested QTRAN seed remains below the collapse cut-off with narrow within-grid variance. This replication covers the five grids and seeds tested; it does not establish invariance outside them. The 6-agent variant on 9×9 preserves VDN's stagnation and QMIX's bi-regime under increased agent count, with QTRAN in full coordination-variance collapse ($\bar{C} < 10^{-3}$) even though its return (0.029) is marginally above zero.

One of the three canonical-grid signatures replicates cross-domain on the RWARE robotic warehouse benchmark [40], a structurally distinct cooperative environment with corridor navigation and sparse reward ($n = 5$ matched training seeds per algorithm). The QTRAN near-zero-dispersion classification recurs under the same numerical rule: seed-mean $\bar{C} \approx 9\times10^{-5}$, mean return 0.000 across all five RWARE seeds, and QTRAN is the only algorithm below the $\bar{C} < 10^{-3}$ collapse threshold. The two non-collapsed algorithms stay above it: QMIX learns the task (mean return 0.86) yet its coordination-variance is compressed to $\bar{C} \approx 3\times10^{-3}$ (only about 3× above the collapse cut-off, unlike its LBF bi-regime), and VDN shows LOW-occupancy STAGNATION ($\bar{C} \approx 0.029$, Low% ≈ 93%), differing from its LBF HIGH-occupancy stagnation. *Both* non-collapsed algorithms therefore sit at low, near-uniform $\bar{C}$ on RWARE, QMIX falling from its LBF $\bar{C} \approx 0.17$. The LBF QMIX-vs-VDN separation therefore shrinks substantially here (Cohen's $d = -1.11$, still a large effect but well below the LBF value), while the collapsed QTRAN class stays sharply separated from QMIX ($d = -2.77$). QTRAN's separation from VDN ($d = -1.23$) is comparable in magnitude to the QMIX-vs-VDN effect, because RWARE compresses VDN's own $\bar{C}$ toward the threshold. Fig. 1 contrasts the recurring QTRAN near-zero-dispersion pattern—below the 10^{-3} threshold in *both* domains—with the domain-dependent VDN/QMIX change. This contrast is associative, not a causal attribution to architecture. A matched-seed episode-runner replication of this four-agent cohort (n=5) shows the same near-zero-dispersion classification, so the observation is not limited to the parallel-runner evaluation. Under a common 100-episode greedy evaluation harness, QTRAN's geometric-mean $\bar{C} \approx 1\times10^{-8}$ with return 0.000 on every seed, whereas VDN does not collapse (geometric-mean $\bar{C} \approx 0.44$, mean return 3.23). Returns are not directly comparable with the parallel-runner figures above: that cohort used the episode runner under a 50-step training time limit rather than the parallel cohort's 500-step limit. Non-collapsed $\bar{C}$ magnitudes are harness-sensitive; only the within-harness QTRAN-versus-VDN contrast is claimed here.

D. Mixer–Optimiser Control

QTRAN was retrained from scratch on the canonical 9×9-5p grid with the *same* optimiser the non-collapsed baselines use—Adam at learning rate 5×10^{-4}. It was evaluated under

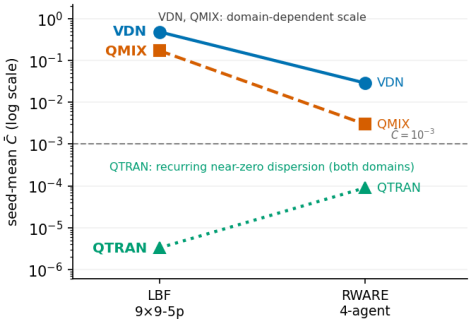


Fig. 1. Seed-mean $\bar{C}$ on LBF 9×9-5p vs. RWARE (log axis; 10^{-3} threshold).

the same 100-episode greedy harness, holding every other setting identical (runner, seeds 0–4, and the 10M-step budget). The optimiser is fixed by the learner implementation rather than by the configuration, so this matched control tests it directly. It does not isolate the mixer: the QTRAN learner also ignores four further keys that the VDN/QMIX learner consumes, and those differences are present here too.

The cohort-mean near-zero-dispersion classification persists. At 10M steps the QTRAN-with-Adam cohort reaches seed-mean $\bar{C} = 6.9\times10^{-4}$, below the $\bar{C} < 10^{-3}$ cut-off and far below both non-collapsed baselines on the same grid: 2.8 orders of magnitude below VDN (0.483) and 2.4 below QMIX (0.172). The regime signature is unchanged from the RMSprop cohort on every seed: Low% = 100, High% = 0, and zero regime transitions per episode. At the 10M-step budget, giving QTRAN the baselines' optimiser does not give it their coordination structure.

The optimiser is nevertheless not inert. The swap raises seed-mean $\bar{C}$ from 3.3×10^{-6} to 6.9×10^{-4}, a factor of about 209, with complete separation between the two cohorts: the lowest Adam seed (2.4×10^{-4}) exceeds the highest RMSprop seed (7.8×10^{-6}) by more than an order of magnitude. One Adam seed reaches 1.6×10^{-3} and thus sits above the cut-off; a checkpoint sweep over the final quarter of training shows two near-cut seeds alternating across the cut-off rather than settling on either side, so their class assignment reflects where training stopped rather than a distinct behavior. The cohort-level verdict is stable under that ambiguity: recomputed at each of the twelve final checkpoints, the seed-mean stays below the cut-off in every case.

The near-zero-dispersion pattern is a *learned* network state, not present in the untrained network: at the earliest logged checkpoint (200 environment steps, effectively untrained) every seed sits above the cut-off ($\bar{C} \approx 2\times10^{-3}$–$1.4\times10^{-2}$), and training drives $\bar{C}$ down by three to four orders of magnitude before the partial recovery just described. The near-zero $\bar{C}$ of this cohort therefore reports a learned numerical state rather than an initialization artifact. One boundary is stated explicitly: the claim is anchored at 10M steps—unlike the RMSprop cohort, whose $\bar{C}$ is flat, the Adam cohort's $\bar{C}$ is still rising at the end of training on all five seeds, so the result bounds behavior at this budget rather than at convergence.

E. Generalization to Unseen Architectures

The taxonomy only partially generalizes beyond the four studied architectures. On three 2025 value-decomposition methods (QFix [9], QTypeMix [8], DuelMIX [7]; five matched seeds each, seeds 0–4, at the same 10M-step budget; Table IX), C separates all 12 successfully-trained (BI-REGIME) seeds from all 3 failed (STAGNATION) seeds, with no seed falling below the $\bar{C} < 10^{-3}$ collapse cut-off. The three failed seeds are the informative case: their zero return is indistinguishable from QTRAN's on the reward curve, yet C separates their numerical patterns: a uniform-high-$\bar{C}$ one-band case ($\bar{C} \in [0.84, 0.99]$) rather than QTRAN-like near-zero dispersion. This difference does not identify the underlying behavioral or causal mechanism. Only two of the three signatures manifest.

F. Held-Out Evaluation on QPLEX

QPLEX is a duplex-dueling architecture not used to design C, choose its thresholds, or fit any free parameter of the diagnostic [4]. Across five QPLEX training seeds on the canonical 9×9-5p-3f-coop LBF grid under the canonical protocol (`time_limit=50`, $t_{\max} = 10$M), test return is 0.000 for every seed at the 10M-step checkpoint (Table IV); that 5/5 return-zero statement is protocol-tied (footnote). The corresponding $\bar{C}$ values are heterogeneous: three exceed 0.7, one is approximately 0.10, and one is 0.036; none falls below the 10^{-3} collapse cut-off. Return collapse and coordination-variance collapse therefore dissociate: QPLEX has zero return on 5/5 seeds without reproducing QTRAN's low-$\bar{C}$ signature. The QPLEX pattern is treated as an architecture-specific heterogeneity mode that may refine the taxonomy, not as evidence for a confirmed fourth signature.*

TABLE IV. HELD-OUT QPLEX PER-SEED $\bar{C}$ AND RETURN (LBF 9×9-5P)

Seed	$\bar{C}$	Return
0	0.956	0.000
1	0.705	0.000
2	0.102	0.000
3	0.036	0.000
4	0.906	0.000

All five seeds fail Prediction H1 under the assumptions stated in the Methodology. H1's sole prediction, operationalized as per-seed $\bar{C} < 10^{-3}$ at 10M steps, does not hold. As the dissociation above shows, the QTRAN-calibrated threshold does not detect every form of policy failure. Per-seed values for the other reported cohorts are available from the corresponding author on reasonable request. The cohort-wide return-collapse at 5/5 is an outcome H1 does not predict and is reported as an observation, not as partial support for H1.

*Per-seed origin: the QPLEX registry contains canonical (`time_limit=50`, $t_{\max} = 10$M) runs for all five seeds plus non-canonical (`time_limit=100`, $t_{\max} = 5$M) runs for seeds 0–4; on the non-canonical sub-cohort 1/5 seeds reached return-collapse and 4/5 retained mid-range returns (0.35–0.59). The 5/5 return-collapse statement therefore applies only to the canonical 10M-step cohort.

V. Discussion

A. Interpretation and Checkpoint Reading

The three signatures support a scope-controlled checkpoint reading (Table VI). The only candidate action evaluated retrospectively is COLLAPSE→consider stopping on canonical LBF. The other mappings are interpretive prompts, not tested interventions. Even the collapse action requires domain-specific validation because it produces a false positive on RWARE and misses QPLEX's return-collapse mode. The constraint–performance ordering is consistent with prior evidence on monotonicity [6], but the present observations do not establish a causal mechanism.

A practitioner should not abort solely because $\bar{C} < 10^{-3}$. On canonical LBF, that one-million-step rule would have saved nine million additional steps for the retained QTRAN seeds without flagging retained QMIX seeds. Before operational use it requires validation on the target domain, algorithm family, Q-value scale, and checkpoint protocol. RWARE false positives, missing VDN early checkpoints, and the absence of an early QPLEX extraction delimit the evidence.

Joint TD-error and final return are evaluated under the matched main protocol. A further post-hoc retained-checkpoint evaluation uses the same 100-episode harness for C, normalized softmax action entropy, the top-1–top-2 Q-gap, pairwise greedy-action disagreement, and within-episode action switching (Table XI). The retained checkpoints are not horizon-matched: VDN is at approximately 7.5M steps, QMIX at 7.5–9.5M, and QTRAN at approximately 10M. Entropy lies near its ceiling for all three cohorts (0.9925–0.9999995), while the Q-gap, disagreement, and switch summaries also distinguish QTRAN from VDN/QMIX. For QTRAN versus QMIX, the absolute seed-level Cohen's d values are 1.86 for C, 2.38 for entropy, 3.71 for Q-gap, 4.47 for disagreement, and 22.35 for switching; at $n = 5$ with unequal checkpoint horizons, these are descriptive point estimates. These associations supply behavioral context but do not show that C measures roles or causes success. Against the two matched main-protocol scalars, C separates canonical QMIX from VDN most strongly at $n = 5$ ($d = -2.93$, versus $+1.60$ for final return and -1.31 for TD-error; Table III). The post-hoc additions are not the published Role Diversity or policy-churn diagnostics, and no superiority over those methods, phase-regime analysis, or neural diversity is claimed.

B. Scope and Transfer

Two descriptive cross-domain observations emerge. QTRAN's near-zero-dispersion pattern appears on LBF, on RWARE under two runners, and under the optimiser control. In contrast, VDN and QMIX magnitudes and occupancy patterns change between LBF and RWARE. These associations are consistent with an architecture-related QTRAN pattern and environment-dependent non-collapse patterns, but the study does not identify the causal contribution of architecture, environment geometry, representation sharing, or reward scale. The LBF/RWARE comparison therefore motivates, rather than proves, a structural explanation and requires domain-specific calibration.

C. Limitations and Reproducibility

The QTRAN near-zero-dispersion claim rests on 25 LBF training seeds across the five 5-agent grids; the underlying per-seed records are available from the corresponding author on reasonable request. The pooled geometric-mean separation of more than four orders of magnitude, on the pool defined above, is the headline; the worst-case two-point extremes ratio reported there is about two orders of magnitude smaller. Layer-1 C targets bounded-reward regimes; Layer-2 $\mathrm{RawVar}(Q)$ is intended where tanh saturates. A further bound is scale identifiability: under monotonic mixing, intermediate cross-architecture $\bar{C}$ magnitudes are read on the realized training scale rather than as scale-invariant distances, while exact zero variance remains zero under finite common rescaling. The finite rule $\bar{C} < 10^{-3}$ is not invariant: the rescaling audit above changes QMIX labels without changing greedy policies.

The canonical QMIX $\bar{C}$ arm is not strictly horizon-matched: its five seeds use last-saved checkpoints (9.50M, 9.50M, 10.40M, 9.75M, and 9.75M), seed 2 interrupted at 10.40M. Omitting it moves $\bar{C}$ from 0.172 to 0.196 and $|d_C|$ versus VDN from 2.93 to 2.71; BI-REGIME and $|d| > 2$ hold either way; no horizon-matched 9–10M checkpoint exists on disk for seed 2, so the mismatch is disclosed rather than re-extracted. H1's literal $\bar{C} \to 0$ prediction does not hold per-seed; the unseen-architecture check is likewise partial; SMACv2 and robotic manipulation were not tested.

Of the 417 training-run attempts in the run index, 349 (84%) ran to at least 95% of their planned timesteps and 36 (9%) stopped before the 10% checkpoint. Because the last group stopped before the 10% checkpoint, no early-checkpoint $\bar{C}$ exists for them, so their exclusion rests on the absence of that checkpoint rather than on a statistical test; the full accounting is recomputed from the retained run index. The Adam control is budget-anchored at 10M steps with cohort-mean reporting. Three of the five seeds sit within a factor of two of the cut-off, and two of them alternate across it between adjacent checkpoints. The QTRAN near-zero-dispersion pattern recurs on all five tested LBF grids, on RWARE, and under the matched Adam control.

Five bounds follow from the design. First, C uses Q-value magnitudes and is not policy- or scale-invariant; tanh bounds the range but does not remove this dependence. Second, occupancy discards order. The separately reported transition rate and seed-0 cross-checkpoint trace in Table X add temporal description, but one retained seed does not validate temporal labels across the cohort. Third, thresholds and the rule form were calibrated on the same VDN/QMIX/QTRAN triad. Sweeping threshold values tests sensitivity, not independent calibration. Fourth, C is not a coordination-quality score: return and dispersion are not monotone, and QPLEX demonstrates return failure without near-zero dispersion. Fifth, joint TD-error and final return are the matched main-protocol comparisons. The post-hoc retained-checkpoint comparison adds softmax action entropy, top-two Q-gap, greedy-action disagreement, and within-episode switching, but its horizons differ and it does not implement Role Diversity or policy churn. All effect sizes use five seeds per cohort and have wide intervals. The additional QTRAN seeds 5–9 were evaluated only after the cut-off was frozen and therefore strengthen same-architecture repeatability, but they are not an independent cal-

ibration set and include no additional VDN/QMIX specificity controls. Artifacts are listed in the Data and Code Availability statement.

VI. Conclusion

This study presented the Coordination Complexity Index (CCI), a checkpoint statistic with no learned parameters that summarizes per-agent greedy-Q dispersion. It separates the reported non-collapsed LBF cohort from the pooled collapsed cohort by more than four orders of magnitude at the geometric mean. On canonical LBF, VDN, QMIX, and QTRAN receive operational one-band, dual-band, and near-zero-dispersion labels. These labels do not by themselves prove stagnation, oscillation, or behavioral roles. QTRAN's pattern recurs on RWARE, while other patterns change across domains. QFix, QTypeMix, and DuelMIX provide partial unseen-architecture evidence. Held-out QPLEX rejects H1 and shows that return collapse need not imply near-zero CCI. The optimiser control shows that the reported QTRAN pattern is learned under the tested setup rather than present at initialization.

Layer-1 C can saturate in reward-unbounded regimes, where Layer-2 $\mathrm{RawVar}(Q)$ is the intended probe. The $\bar{C} < 10^{-3}$ rule is only a retrospectively supported candidate on retained canonical-LBF checkpoints: it has incomplete early-checkpoint coverage and produces a false positive on RWARE. It should therefore prompt targeted inspection or a domain-validated stopping policy, not automatic termination. With the threshold frozen, the additional QTRAN seed extension retains 5/5 positives at 10% and 5/5 near completion, but this one-class repeatability result does not remove the need for independent calibration and negative controls.

Acknowledgments

This research was conducted at the Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok. The authors thank the developers of the EPyMARL framework [34], the LBF environment [39], and the RWARE environment [40].

Funding

The authors received no specific external grant from public, commercial, or non-profit funding agencies; all experiments were run on the lead author's own workstation.

Conflicts of Interest

The authors declare no competing interests.

Data and Code Availability

The analysis code, run index, matched seed matrix, threshold and scale-sensitivity sweeps, the held-out QTRAN per-seed evaluation records, and the raw Sacred metrics and configurations used for the reported results are held by the authors and are available from the corresponding author on reasonable request. A permanent public repository and archival DOI will be added after deposit; no inactive URL is asserted here. Reported table values of C and the `macci` implementation both use the unbiased variance estimator ($\mathrm{ddof}=1$), as defined with Eq. (2).

Declaration on Generative AI Tools

During the preparation of this work the authors used Claude (Anthropic) in the writing process, to draft and revise the text, and in the research process, where it assisted in writing the analysis scripts that recompute the reported statistics and generate the figures from logged run output. The learning algorithms are the framework's own implementations, the `macci` implementation is the authors' own work, and the study design, the experiments, and the claims and conclusions are the authors' own; no reported quantity was produced by the model. Every number in this article is derived from the training runs' own logged output, retained checkpoint evaluations, or the run index built over them; the analysis scripts and supporting records are available as stated in the Data and Code Availability statement. The authors reviewed and edited every AI-assisted output and take full responsibility for the content of the published article.

References

[1] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2018, preprint arXiv:1706.05296. [Online]. Available: https://dl.acm.org/doi/10.5555/3237383.3238080

[2] T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[3] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[4] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "QPLEX: Duplex dueling multi-agent Q-learning," in *International Conference on Learning Representations (ICLR)*, 2021.

[5] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, "Loss of plasticity in deep continual learning," *Nature*, vol. 632, no. 8026, pp. 768–774, 2024.

[6] J. Hu, S. Jiang, S. A. Harding, H. Wu, and S.-w. Liao, "Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning," *arXiv preprint arXiv:2102.03479*, 2023, ICLR 2023 Blog Post.

[7] E. Marchesini, A. Baisero, R. Bhati, and C. Amato, "On stateful value factorization in multi-agent reinforcement learning," in *Proceedings of the 24th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2025, preprint arXiv:2408.15381. [Online]. Available: https://dl.acm.org/doi/10.5555/3709347.3743778

[8] S. Fu, S. Zhao, T. Li, and Y. Yan, "QTypeMix: Enhancing multi-agent cooperative strategies through heterogeneous and homogeneous value decomposition," *Neural Networks*, vol. 184, p. 107093, 2025.

[9] A. Baisero, R. Bhati, S. Liu, A. Pillai, and C. Amato, "Fixing incomplete value function decomposition for multi-agent reinforcement learning," *arXiv preprint arXiv:2505.10484*, 2025.

[10] H. Li, H. Zhou, Y. Zou, D. Yu, and T. Lan, "ConcaveQ: Non-monotonic value function factorization via concave representations in deep multi-agent reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024.

[11] K. Liu, T. Zhang, X. Xu, and Y. Zhao, "Counterfactual value decomposition for cooperative multi-agent reinforcement learning," *Neural Networks*, 2025.

[12] D. Yao, C. Sun, K. Li, K. Zhou, H. Li, R. Wang, and L. Liu, "CMIX: Causal value decomposition for cooperative multi-agent reinforcement learning," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2024.

[13] A. Huang, Y. Wang, R. Liu, H. Zou, and X. Zhou, "QVF: Incorporating quantile value function factorization into cooperative multi-agent reinforcement learning," *Pattern Recognition*, 2025.

[14] Z. Liu, L. Wan, S. Sun, X. Sui, X. Chen, X. Lan, and N. Zheng, "Enhancing value decomposition with target transformation in cooperative multi-agent reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[15] S. Wang, R. Zhang, Y. Zhou, J. Shao, and Y. Cheng, "QuadQ: Quadratic-based value decomposition for cooperative policy optimization in multi-agent reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, 2026.

[16] W. Fu, C. Yu, Z. Xu, J. Yang, and Y. Wu, "Revisiting some common practices in cooperative multi-agent reinforcement learning," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, vol. 162, 2022, pp. 6863–6877.

[17] L. Wan, Z. Liu, X. Chen, X. Lan, and N. Zheng, "Greedy based value representation for optimal coordination in multi-agent reinforcement learning," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, vol. 162, 2022, pp. 22 512–22 535.

[18] G. Papadopoulos, A. Kontogiannis, F. Papadopoulou, C. Poulianou, I. Koumentis, and G. Vouros, "An extended benchmarking of multi-agent reinforcement learning algorithms in complex fully cooperative tasks," in *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2025, pp. 1613–1622.

[19] R. Gorsane, O. Mahjoub, R. J. de Kock, R. Dubb, S. Singh, and A. Pretorius, "Towards a standardised performance evaluation protocol for cooperative MARL," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[20] B. Ellis, J. Cook, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson, "SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*, 2023.

[21] S. Hu, C. Xie, X. Liang, and X. Chang, "Policy diagnosis via measuring role diversity in cooperative multi-agent RL," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 162, 2022.

[22] M. Bettini, A. Shankar, and A. Prorok, "System neural diversity: Measuring behavioral heterogeneity in multi-agent learning," *Journal of Machine Learning Research*, vol. 26, no. 163, pp. 1–27, 2025.

[23] T. Wang, L. Zeng, W. Dong, Q. Yang, Y. Yu, and C. Zhang, "Context-aware sparse deep coordination graphs," in *International Conference on Learning Representations (ICLR)*, 2022, arXiv:2106.02886.

[24] E. Mieczkowski, R. Mon-Williams, N. Bramley, C. G. Lucas, N. Vélez, and T. L. Griffiths, "Predicting multi-agent specialization via task parallelizability," *arXiv preprint arXiv:2503.15703*, 2025.

[25] T. Verma and P. Varakantham, "Learning individual policies in large multi-agent systems through local variance minimization," *arXiv preprint arXiv:2212.13379*, 2022.

[26] W. Khlifi, S. Singh, O. Mahjoub, R. de Kock, A. Vall, R. Gorsane, and A. Pretorius, "On diagnostics for understanding agent training behaviour in cooperative MARL," in *XAI4DRL Workshop at the AAAI Conference on Artificial Intelligence*, 2023, arXiv:2312.08468.

[27] G. Sokar, R. Agarwal, P. S. Castro, and U. Evci, "The dormant neuron phenomenon in deep reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, vol. 202, 2023, pp. 32 145–32 168.

[28] H. Qin, C. Ma, M. Deng, Z. Liu, S. Mei, X. Liu, C. Wang, and S. Shen, "The dormant neuron phenomenon in multi-agent reinforcement learning value factorization," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 35 727–35 759.

[29] Z. Liu, Z. Gao, H. Qin, J. Hu, J. Wu, M. Zhu, H. Zhang, C. Ma, S. Shen, and C. Wang, "Stagnant neuron: Towards understanding the plasticity loss in multi-agent reinforcement learning value factorization methods," *arXiv preprint arXiv:2606.25335*, 2026.

[30] T. Schaul, A. Barreto, J. Quan, and G. Ostrovski, "The phenomenon of policy churn," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 2537–2549.

[31] Y.-N. Weng and H.-W. Lee, "How exploration breaks cooperation in shared-policy multi-agent reinforcement learning," *arXiv preprint arXiv:2601.05509*, 2026.

[32] A. Yamaguchi, "Emergent coordination and phase structure in independent multi-agent reinforcement learning," *arXiv preprint arXiv:2511.23315*, 2025.

[33] M. A. Cardei, M. Landers, and A. Doryab, "Coordination matters: Evaluation of cooperative multi-agent reinforcement learning," *arXiv preprint arXiv:2605.06557*, 2026.

[34] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," in *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[35] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[37] T. Tieleman and G. Hinton, "Lecture 6.5—RMSProp: Divide the gradient by a running average of its recent magnitude," 2012. [Online]. Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[38] T. Hu, Y. Cui, R. Tang, B. Luo, and K. Li, "Beyond monotonicity: Revisiting factorization principles in multi-agent Q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 26, 2026, pp. 21 876–21 884.

[39] S. V. Albrecht and S. Ramamoorthy, "A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems," in *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013.

[40] F. Christianos, L. Schäfer, and S. V. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[41] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[42] L. V. Hedges, "Distribution theory for Glass's estimator of effect size and related estimators," *Journal of Educational Statistics*, vol. 6, no. 2, pp. 107–128, 1981.

[43] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. G. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

APPENDIX

APPENDIX A
TABLES REFERENCED IN THE TEXT

This appendix collects the configuration and per-seed tables the preceding sections refer to, so that every value cited in the text can be read from this article alone.

Table V lists the shared training hyperparameters. They are applied identically to VDN/QMIX/QTRAN on the LBF grids except the optimiser (Adam for VDN/QMIX, RMSprop for QTRAN; learning rate 5×10^{-4} throughout), so that the mixer is the primary intended per-algorithm difference; the RWARE cohorts run to 2×10^7 steps.

TABLE V. SHARED TRAINING HYPERPARAMETERS

Hyperparameter	Value
Learning rate	5×10^{-4} (all)
Optimiser	Adam for VDN and QMIX; RMSprop (α=0.99) for QTRAN[b]
Gradient-norm clip	10
Discount γ	0.99
TD(λ)	0.5
Double-Q	True
Batch size	128 episodes
Replay-buffer size	5000 episodes
Target-update interval	200 (hard)
ϵ-greedy schedule	$1.0 \rightarrow 0.05$ over 5×10^5 steps
Agent network	Feed-forward MLP, hidden dim 64 (`use_rnn`=False)
Agent input	observation + one-hot agent id (`obs_agent_id`=True)
Runner / parallel envs	parallel, `batch_size_run`=4[a]
Training horizon $t_{\max}$	1×10^7
Mixer (intended difference)	vdn / qmix / qtran_base

[a]The static configuration files differ on the runner key (QTRAN declares `parallel`/`batch_size_run`=4 while QMIX and VDN declare `episode`/1); the launch scripts override all three to `parallel`, `batch_size_run`=4 on the command line, so the realized runs are matched, as confirmed by the run inventory. One mixer-internal width is not numerically identical (`mixing_embed_dim`: QMIX 32, QTRAN 64, VDN n/a); this is treated as part of the mixing architecture under test rather than as a shared hyperparameter.
[b]The optimiser is hardcoded per learner rather than exposed as a configuration key, so it could not be matched by the shared-configuration discipline: `q_learner` (VDN, QMIX) constructs Adam at lr=5×10^{-4}, and `qtran_learner` constructs RMSprop at the same learning rate with α=0.99, ϵ=10^{-5}. Those two RMSprop-specific constants are never read on the Adam path. This is the difference the QTRAN-with-Adam control in the main text isolates.

Table VI states the scope-controlled reading of CCI. Its criteria reproduce the classifier of Section III-C and are evaluated in order. Signature names are operational labels; transitions are descriptive and do not enter classification.

TABLE VI. SCOPE-CONTROLLED CCI INTERPRETATION

Operational label	Criterion	Interpretation / next check
Collapse (first)	$\bar{C} < 10^{-3}$	Consider stopping only after target-domain validation; otherwise inspect return, scale, and later checkpoints.
Uni-regime (STAGNATION)	$\max(\text{High}\%, \text{Low}\%) > 80\%$	Inspect additional checkpoints; one-band occupancy alone does not establish halted adaptation.
Dual-occupancy (BI-REGIME)	otherwise	Inspect transition traces or additional checkpoints; occupancy alone does not establish oscillation.

Table VII reports the per-seed seed-mean $\bar{C}$ at the retained early-budget checkpoint. Values are extracted from frozen checkpoints under 50 or 100 greedy episodes: 50 for the five QTRAN seeds and for QMIX seeds 0 and 1, 100 for the remaining eight cells. The collapse threshold 10^{-3} is the one used in the signature-classification rule of main study Section III-C.

TABLE VII. PER-SEED SEED-MEAN $\bar{C}$ AT THE RETAINED EARLY-BUDGET CHECKPOINT (LBF $9 \times$ 9-5P-3F-COOP-V3)

Seed	QMIX (10%)	QTRAN (10%)	QPLEX (100%)
$s = 0$	0.045	2.4×10^{-6}	0.956
$s = 1$	0.163	2.3×10^{-6}	0.705
$s = 2$	0.043	2.4×10^{-6}	0.102
$s = 3$	0.034	1.9×10^{-6}	0.036
$s = 4$	0.497	1.8×10^{-6}	0.906
Range	$[0.034, 0.497]$	$\sim 10^{-6}$	$[0.036, 0.956]$
Return-collapse?	NO	YES (5/5)	**YES (5/5, return = 0)**
C-collapse? (vs 10^{-3})	NO	YES (5/5)	**NO (0/5)**

Table IX extends the signature taxonomy to 2025 value-decomposition contemporaries on LBF $9 \times$ 9-5p ($n = 5$ each, agent-only C extraction). "Working" = mean return > 0; "Failed" $= 0$. The DuelMIX working subset reproduces the prior $\bar{C} = \{0.122, 0.144, 0.063\}$ exactly; no contemporary seed exhibits QTRAN-style COLLAPSE ($\bar{C} < 10^{-3}$).

TABLE VIII. PER-GRID $\bar{C}$ AND RETURNS, QMIX VS. VDN, ACROSS LBF GRID SIZES AND A SIX-AGENT VARIANT

Grid	V-$\bar{C}$	Q-$\bar{C}$	d_C/g_C	VDN Ret.	QMIX Ret.	P
8×8	0.510	0.083	$-10.10/-9.12$	0.678 [0.61, 0.74]	0.775 [0.71, 0.84]	0.99
9×9	0.483	0.172	$-2.93/-2.65$	0.632 [0.55, 0.71]	0.732 [0.70, 0.76]	0.97
10×10	0.477	0.208	$-1.47/-1.33$	0.590 [0.54, 0.65]	0.759 [0.73, 0.80]	0.92
11×11	0.475	0.099	$-10.08/-9.10$	0.567 [0.49, 0.66]	0.657 [0.59, 0.70]	1.00
12×12	0.487	0.090	$-12.78/-11.54$	0.504 [0.47, 0.54]	0.631 [0.58, 0.67]	1.00
9×9-6p	0.493	0.220	$-1.55/-1.40$	0.737 [0.67, 0.78]	0.843 [0.83, 0.86]	1.00

TABLE IX. SIGNATURE TAXONOMY EXTENDED TO 2025 VALUE-DECOMPOSITION CONTEMPORARIES ON LBF 9×9-5P

Method (2025)	Working	$\bar{C}$ (working)	Failed	$\bar{C}$ (failed)
QFix (Baisero 2025)	4 BI-REGIME	0.04–0.12	1 STAGNATION	0.99
QTypeMix (Fu 2025)	5 BI-REGIME	0.03–0.09	0	—
DuelMIX (Marchesini 2025)	3 BI-REGIME	0.06–0.14	2 STAGNATION	0.84–0.99

Table X gives a descriptive seed-0 cross-checkpoint trace from retained canonical-LBF checkpoints (50 greedy episodes per checkpoint). The trace preserves checkpoint order but is not a cohort-level temporal classifier or proof of recovery, stagnation, or continued adaptation.

TABLE X. DESCRIPTIVE SEED-0 CROSS-CHECKPOINT TRACE FROM RETAINED CANONICAL-LBF CHECKPOINTS

Algorithm	Retained steps (M)	$\bar{C}$ sequence	Transitions/episode	Permitted inference
VDN	0.0002, 2.50, 5.00, 7.50 1.00, 2.00, 4.00,	0.0049, 0.450, 0.565, 0.531 0.045, 0.039, 0.065,	0, 6.60, 0.14, 0.14 4.24, 4.66, 3.18,	Single seed: late one-band persistence.
QMIX	6.00, 8.00, 9.50 1.00, 2.00, 4.00,	0.084, 0.094, 0.094	2.98, 2.38, 2.68	Single seed: repeated within-episode crossings.
QTRAN	6.00, 8.00, 9.95	3.4×10^{-7}–7.8×10^{-6}	0 at all six checkpoints	Single seed: persistent near-zero dispersion.

Table XI compares checkpoint diagnostics post hoc on retained canonical-LBF runs (five seeds per algorithm, 100 greedy episodes per seed). Entropy is $H(\mathrm{softmax}(Q))/\log|\mathcal{A}|$ at temperature 1; Q-gap is the per-agent top-1 minus top-2 action-value gap; disagreement is the fraction of agent pairs choosing different greedy actions; switch is the within-episode fraction of agents whose greedy action changes between consecutive timesteps. These two behavioral summaries are not Role Diversity or policy churn.

TABLE XI. POST-HOC COMPARISON OF CHECKPOINT DIAGNOSTICS ON RETAINED CANONICAL-LBF RUNS

Algorithm	$\bar{C}$	Entropy	Q-gap	Disagreement	Switch
VDN	0.480	0.9925	0.1058	0.770	0.522
QMIX	0.181	0.9991	0.0369	0.741	0.494
QTRAN	3.31×10^{-6}	0.9999995	0.00123	0.472	0.0092

Table XII reports the frozen-cut-off extension on five additional canonical-LBF QTRAN training seeds. Each checkpoint is evaluated for 100 greedy episodes using the pre-existing $\bar{C} < 10^{-3}$ cut-off. This is a positive-class repeatability check, not an independent calibration or specificity set.

TABLE XII. FROZEN-CUT-OFF EXTENSION ON FIVE ADDITIONAL CANONICAL-LBF QTRAN TRAINING SEEDS

Seed	Early step	Early $\bar{C}$	Early return	Final step	Final $\bar{C}$	Final return
5	1,000,200	1.44×10^{-6}	0.000	10,000,790	9.99×10^{-6}	0.007
6	1,000,200	1.35×10^{-5}	0.003	10,000,977	5.74×10^{-6}	0.000
7	1,000,200	1.67×10^{-6}	0.003	10,000,600	2.65×10^{-6}	0.010
8	1,000,200	1.98×10^{-6}	0.000	10,000,200	2.21×10^{-8}	0.000
9	1,000,200	1.01×10^{-6}	0.000	10,000,200	7.93×10^{-6}	0.013

Predicting Crowd Agitation from Raw Video: A Lightweight Multimodal Regression Framework for Social Tension Estimation

Abderrahim Ouza[1], Khalid Ounachad[2], Mohamed El Ghmary[3], Ali Choukri[4]
Faculty of Science, Ibn Tofail University, Kenitra, Morocco[1,4]
USMS Sultan Moulay Slimane University, Beni Mellal, Morocco[2]
ENSAM, Mohammed V University in Rabat, Morocco[3]

***Abstract*—Continuous, quantitative awareness of crowd tension – a perceptually grounded, continuous measure of crowd agitation, distinct from a binary violence label, an unnormalized abnormal-event score, or raw motion magnitude – is required for proportionate public-safety response; however, the dominant paradigm in computer vision approaches remains restricted to post-hoc binary violence detection. To overcome this limitation, an edge-centric multimodal regression framework is proposed that estimates normalized Social Tension Index ($\mathrm{STI} \in [0, 1]$) directly from surveillance video without manual feature engineering. A frozen CLIP encoder (512-d) captures spatial semantics, while Farnebäck optical-flow descriptors (2-d) model short-range dynamics and audio RMS energy (1-d), where available, provides acoustic agitation; this 515-dimensional representation is mapped to a tension score by a gradient-boosted LightGBM regressor. A hybrid annotation protocol utilizing expert scoring together with motion and audio heuristics has high agreement between annotators ($\kappa = 0.81$), and a label-integrity study demonstrates no circular bias introduced by heuristic components (only a 0.031 R^2 drop under purely human labels). Five-fold cross-validation on RLVS gives $\mathrm{MAE} = 0.0255 \pm 0.0053$ and $R^2 = 0.8732 \pm 0.0631$, with a binary accuracy of 93.75% downstream (precision 100%/recall 87.5% on High-tension alerts). Measured on a single CPU core, the edge decision head runs in 1.65 ms per clip (606 clips/s); the frozen-encoder design does not require any retraining on specific scenes, and this measured feasibility on resource-restricted hardware makes the framework a promising tool for graduated territorial risk monitoring, pending field validation.**

***Keywords*—Social tension index; multimodal video analysis; deep visual embeddings; optical-flow motion dynamics; violence detection; real-time surveillance; LightGBM; CLIP; crowd behavior; territorial risk monitoring**

I. Introduction

Effective territorial administration and public safety management increasingly demands real-time situational awareness of social dynamics in urban and peri-urban spaces. The rapid escalation from ordinary crowd activity to collective agitation, visible through subtle shifts in movement patterns, crowd density, and acoustic energy, can precede organized unrest by several minutes, a window during which timely intervention could prevent harm and reduce the cost of crisis response [3]. For this reason, detecting and quantifying this precursor *tension state* – a continuous, perceptually grounded measure of crowd agitation, distinguished from a binary violence label by its support for partial and escalating states, and from an unnormalized abnormal-event score or raw motion magnitude by being calibrated to human perceptual judgment of escalation risk rather than a fixed event category – rather than reacting to overt violence after being committed, is strategically critical for urban governance bodies, civil protection authorities, and smart-city management platforms.

Traditional alert systems that rely on fixed thresholds or rules set by experts are limited because they depend on collated, retrospective reports and delayed dissemination; this is a crucial limitation in fast-paced public-space scenarios [4]. Camera-based surveillance infrastructure is already ubiquitous in the urban environment and provides a rich perceptual baseline for high-fidelity crowd monitoring; however, the computer vision algorithms that are generally used to consider these feeds remain trapped in a binary paradigm [5]. Each clip is classified as either violent or non-violent, and an alert is raised after violence has been detected [23]. But this is a hard binary that throws away detailed intermediate information about *how tense* a scene is, and impedes the graduated territorial risk management response that proportionate intervention requires. A surveillance operator getting a binary "violent" flag has no history of the previous tension build-up, no sense of how fast this escalation occurred, and no numerical basis to decide if they should engage personally, but alert their team just for moral support, or intervene actively. Binary detection is thus inherently a post-hoc, high-latency decision that offers no signal during the escalation preceding overt violence, whereas continuous social-tension estimation treats each clip as one sample of a time-varying, real-valued process that can support trend analysis and multi-level alerting.

A principled alternative is risk scoring on a continuous basis. Similarly, a time-varying tension index with real-valued output can give ground for time-series analysis of trends and escalation rates, allowing for multi-level alert protocols (as is the case with seismic-hazard monitoring, which provides continuous probabilities rather than binary earthquake/no-earthquake labels [19]). Such indicators are typically used in the crisis-prediction literature, but none prior have proposed a video-based framework providing continuous, normalized tension-scores that are real-time, multimodal, and can be deployed over non-specialized hardware without scene-specific retraining. This work is guided by three research questions: (RQ1) does optical flow provide information beyond visual semantic features; (RQ2) does audio improve prediction when available; and (RQ3) can the frozen encoder generalize across

scenes without scene-specific retraining? These are addressed respectively by the modality ablation (Section V-G), the feature-correlation analysis (Section V-H), and the encoder comparison and cross-validation stability results in Section V.

To address this gap, a machine-learning-based framework is presented that transforms real-world video, without manual feature engineering or scene-specific fine-tuning, into a continuous-valued Social Tension Index (STI). The STI is normalized to be in the range $[0, 1]$ and obtained from three complementary modalities: frozen CLIP visual semantics, motion dynamics from dense optical flow [2], and optional acoustic energy calculated from the audio track. A gradient-boosted regressor, LightGBM [15], maps the 515-dimensional fused feature vector to a tension score, with a small memory footprint and a measured 1.65 ms edge inference latency (after offline CLIP feature extraction), making it suitable for deployment onto off-the-shelf resource-constrained edge surveillance hardware without the need for a GPU alongside it.

The principal contributions of this work are as follows:

- A continuous Social Tension Index (STI$\in[0, 1]$) computed directly from raw video; it supersedes the binary violence labels with a continuous-valued measure of crowd agitation to enable early-warning detection and temporal trend analysis.
- A lightweight multimodal feature pipeline consisting of frozen CLIP-ViT-B/32 [1] spatial embeddings, Farnebäck dense optical-flow statistics [2], optional audio RMS energy, and no scene-specific fine-tuning, distilled into a compact 515-dimensional representation.
- A hybrid annotation protocol integrating expert human scoring with quantitative motion and audio heuristics, validated by strong inter-annotator agreement ($\kappa = 0.81$) and accompanied by a label-integrity analysis that rules out circular bias between heuristic label components and model input features.
- RLVS public benchmark experimental validation, featuring an ablation study comparison of different feature-extractors, modality-classification analysis, and bootstrap confidence interval evaluation, showing competitive performance and excellent generalization (R^2=0.8732$\pm$0.0631); cross-dataset evaluations using RWF-2000 and HockeyFight are left for future work.
- A publicly available open-source implementation, distributed as a Colab notebook, which can be easily tailored to novel surveillance scenarios without dataset-specific retraining.

The rest of this study is structured as follows. Section II surveys related work. Section III details the proposed framework. Section IV describes datasets and preprocessing. Section V presents comprehensive experimental results. Section VI discusses the results. Section VII discusses limitations. Section IX concludes.

II. Related Work

A. Violence and Abnormal-Event Detection in Surveillance Video

Early works of event understanding in the video-based violence detection field were based on hand-crafted motion descriptors such as optical flow histograms and spatio-temporal interest points for classification of aggressive vs. benign motion patterns [7]. These skeleton-based methods then took advantage of pose-estimation graphs to geometrically represent the structural relation of joints in the body and subsequently showed outstanding results on normal single-person controlled scenarios [4]. 3D convolutional networks were proposed to mimic human perception, simultaneously modeling space and time data with a large improvement in accuracy on popular benchmarks such as RLVS [8], [16], and a dedicated deep architecture for violence detection in surveillance footage further improved on this line of work [22]. Transformer architectures with multihead attention have been designed in more recent years to cover long-range temporal dependencies across video clips and achieve state-of-the-art performance on standard violence benchmarks [11], [16]. More recently, dual-path motion encoding strategies that separate short- and long-range motion cues have been proposed to improve the robustness of binary violence classifiers under varying crowd densities [24]. Yet, a fundamental limitation of all these existing methods is that they generate binary labels and are primarily offline post-hoc algorithms, which preclude continuous monitoring of tension for real-time predictive analytics.

B. Crowd Analysis, Optical-Flow Modeling, and Anomaly Detection

Crowd analysis is a broader umbrella of tasks, including density estimation, trajectory prediction, flow modeling, and anomaly detection [17]. Multi-scale crowd-feature frameworks grounded in statistical-mechanics principles model crowd motion fields directly from dense optical flow, treating flow magnitude and disorder as proxies for collective crowd state [17]. Such anomaly detection algorithms typically learn a model of normal crowd behavior, and treat deviations from this described model as anomalies, producing scalar anomaly scores without any semantics about social tension [18]. The use of self-supervised methods has demonstrated that crowd representations can be learned in an unsupervised manner and generalized to new environments with minimal annotations [9]. Works extending this to blend motion data with multimodal signals (shouting, breaking glass) or body-pose cues have shown that cross-modal fusion can achieve robustness in challenging capture conditions [6], [20]. Several two-stream neural network architectures that combine appearance and motion streams have been very successful in real-time surveillance with applications for anomaly detection [16]. A smaller, related body of work targets the affective or agitation state of a crowd directly, by combining crowd-behavior descriptors with classification heads tuned to intermediate severity levels [8]; such systems, however, still report categorical severity levels rather than a normalized, continuous index. Yet, the existing literature has not yet specifically covered the problem of continuous, normalized tension estimation on heterogeneous public datasets and deployment environments.

C. Foundation Models and Vision-Language Models in Video Understanding

Contrastive learning between visual and textual representations during large-scale (400 million image-text pairs) pre-training enables the model [1] to obtain rich scene-level embeddings that generalize well. Recent works have shown that CLIP features transfer well to surveillance and event-understanding tasks with little-to-no fine-tuning [10], whilst the Vision Transformer (ViT) backbone [21] on which CLIP is built, provides strong spatial scene representations of complex crowd scenes. By making use of lightweight decision models (such as gradient-boosted trees and linear regressors) on top of frozen foundation-model embeddings, training cost and inference latency is reduced significantly compared to end-to-end fine-tuning approaches [13], [14]. This paradigm is particularly relevant in surveillance contexts where annotated data is sparsely available, scalable annotation is governed by several privacy constraints and memory-limited edge hardware typically cannot accommodate full computational budgets. Outside of violence detection explicitly, other dense video-understanding tasks have seen lightweight multi-scale architectures with such edge-hardware constraints, such as procedures for human parsing that involve combining dilated spatial pyramid pooling with attention-guided decoders both to reduce parameter counts while preserving segmentation accuracy [26]–[28].

D. Continuous and Multimodal Risk Scoring in Surveillance

Replacing Binary Event Detection with Continuous Risk Estimation. This has been studied in related fields. Autoencoder-based temporal regularity methods have realised that continuous anomaly score trajectory can be obtained smoothly without class-specific labels from raw surveillance video [19]. In anomaly detection, Since these methods are all based on reconstruction error, they inherently produce continuous scores but these are azimuthally unnormalized, and not semantically calibrated to human perceptual judgments [3], [5]. In the broader crisis-prediction literature, composite indicators combining socioeconomic, geospatial, and media signals are typically employed in forecasting macro signals of territorial instability and proactive-governance frameworks [25], but operate while working with day-to-week temporal granularity as opposed to clip or frame level. Prior audio-visual fusion work for violence detection has also shown that combining acoustic and visual streams improves robustness [20], but remains oriented toward binary event labels. To the best of available knowledge, no prior work combines continuous normalized tension scoring with real-time multimodal video analysis at clip-level granularity.

E. Research Gap

Existing video-based surveillance systems address binary violence detection (offline, high-latency), anomaly detection (scalar, semantically uncalibrated), or geopolitical forecasting (macro-level, not video-based). Even the closest prior systems that fuse motion, audio, or pose cues for crowd or violence classification [8], [20] output a categorical or unnormalized score rather than a continuous index tested for circularity against its own heuristic label components – this, rather than multimodal fusion itself, is the genuinely new element of the present work. The combination of a continuous normalized tension score, a lightweight multimodal regression head, an explicit label-integrity analysis, and real-time deployability on standard hardware has not been explicitly proposed for video-based urban surveillance. This work fills that gap by offering a unified regression framework calibrated to produce an interpretable, video-derived STI that integrates directly into existing surveillance infrastructures without scene-specific retraining.

III. Proposed Framework

A. System Overview

Fig. 1 illustrates the end-to-end STI estimation pipeline. Given a short video clip of 8–16 frames, three parallel feature-extraction branches capture complementary aspects of the scene: 1) a frozen CLIP visual encoder provides high-dimensional spatial semantics encoding crowd appearance and scene context; 2) a Farnebäck optical-flow module quantifies the intensity and disorder of crowd motion between consecutive frames; and 3) an optional audio extractor measures acoustic agitation energy where synchronized audio is available. After normalizing the outputs of all the branches, a 515-dim feature vector is produced and given to train a gradient-boosted regressor using LightGBM that outputs a continuous score in $[0, 1]$. During inference, a hard-threshold $\tau = 0.5$ (Section III-H) is used to convert it back into binary Low/Normal or High tension for downstream alert generation.

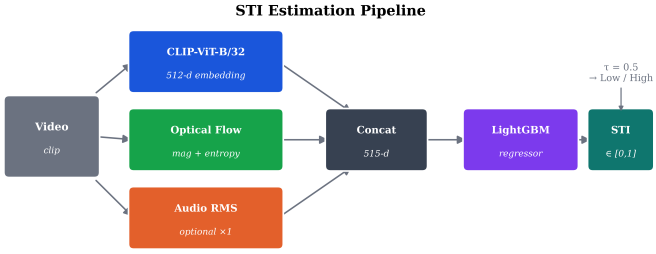


Fig. 1. Proposed multimodal STI estimation pipeline for continuous social tension prediction from video clips.

B. Spatial Feature Extraction via Frozen CLIP

Spatial context is encoded using the CLIP-ViT-B/32 model [1], pretrained on 400 million image-text pairs via contrastive alignment between visual and language representations. For each clip, $N_f = 8$ frames are uniformly sampled from the full clip duration, resized to 224×224 pixels, and normalized using the CLIP ImageNet statistics ($\mu = [0.481, 0.457, 0.408]$, $\sigma = [0.269, 0.261, 0.276]$). Each frame is passed through the frozen visual encoder to produce a 512-dimensional unit hypersphere embedding. The per-frame embeddings are mean-pooled to form a single clip-level spatial descriptor:

$$\mathbf{f}_{\text{clip}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \phi_{\text{CLIP}}(\mathbf{I}_i) \in \mathbb{R}^{512}, \tag{1}$$

where, $\phi_{\text{CLIP}}(\cdot)$ denotes the frozen encoder. Temporal mean-pooling is motivated by the clip-level nature of the

tension score: a holistic representation of the scene state over the entire clip is sought, not per-frame classification. Using a completely frozen encoder eliminates the risk of overfitting on the limited surveillance annotation budget while exploiting the broad scene understanding capabilities acquired during large-scale vision language pretraining.

C. Motion Feature Extraction via Optical Flow

Short-range motion dynamics are captured using Farnebäck dense optical flow [2] computed on consecutive grayscale frame pairs. For each consecutive pair (F_t, F_{t+1}), a dense displacement field $\mathbf{V}(x, y) = (u, v)$ is estimated, where (u, v) denotes the horizontal and vertical pixel displacement at location (x, y). The Farnebäck algorithm uses polynomial expansion of the local signal with a Gaussian window (neighborhood size 5, $\sigma = 1.1$) and three pyramid levels to handle large displacements in high-density crowd scenes. Two scalar statistics are derived from the per-pixel magnitude field $M(x, y) = \sqrt{u^2 + v^2}$:

(i) Mean flow magnitude $\bar{m} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} M(x, y)$, where Ω is the set of all valid pixel locations. This statistic measures the overall displacement intensity and serves as a direct proxy for the scale and speed of crowd movement.

(ii) Normalized flow entropy

$$H = -\frac{1}{\log_2 K} \sum_{k=1}^{K} p_k \log_2 p_k, \tag{2}$$

where, p_k is the fraction of magnitude values falling in bin k of a $K = 32$-bin histogram. $H \in [0, 1]$ is normalized by the maximum achievable entropy $\log_2 K$. High entropy indicates disordered, chaotic motion distributed across many magnitude bins, a key perceptual marker of crowd panic, confrontation, or dispersal. Together, the two dimensional motion descriptor is $\mathbf{f}_{\text{flow}} = [\bar{m},\ H]^\top \in \mathbb{R}^2$.

D. Audio Feature Extraction

For audio tracks that are available and temporally aligned with the video, the clip is downsampled to 16 kHz mono and root-mean-square (RMS) amplitude computed over the entire segment:

$$f_{\text{audio}} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} s(t)^2}, \tag{3}$$

where, $s(t)$ represents the audio sample at time step t, and T is the number of total samples. RMS energy captures shouting, crowd noise, and other agitation-related acoustic signatures without needing complicated spectral decomposition. If audio is missing, or if it does not pass a synchronization quality test (RMS ¡ 0.01 on normalized samples, or an audio track that is shorter than 80% of the video), $f_{\text{audio}} = 0$ is set to keep the spatial and motion branches unaffected. In practice, a large proportion of RLVS clips obtained from the Internet do not have any aligned audio; however, for these, the model simplifies to CLIP+Motion. This contribution from the audio branch reported in the ablation study (Section V-G) thus constitutes a conservative lower bound on its utility in real-world deployments with an uncontrolled background of audio.

E. Feature Fusion and Normalization

Independent of each other, the scalar features are normalised to all fall in $[-1, 1]$ and then concatenated together between the three branches.

$$\mathbf{x} = \left[\mathbf{f}_{\text{clip}};\ \hat{m}_{\text{norm}};\ \hat{H}_{\text{norm}};\ \hat{f}_{\text{audio,norm}}\right] \in \mathbb{R}^{515}. \tag{4}$$

Normalization statistics (min and max for each scalar feature) are fit only on the training split, then applied identically to the validation and test splits to avoid leakage. The 512-dimensional CLIP branch is passed through without any additional normalization, as the unit-hypersphere constraint of the CLIP encoder already leads to embeddings that are bounded and comparable in magnitude across all clips.

F. STI Label Construction and Label Integrity Analysis

Ground-truth STI labels are constructed using a combined annotation protocol of human perceptual scoring and quantitative motion and audio heuristics. Three independent annotators, trained research assistants with backgrounds in social science and computer vision, scored each clip on a continuous scale $[0, 1]$ according to three criteria: 1) visible aggression level and confrontation intensity; 2) crowd density, motion speed, and directional disorder; and 3) contextual signals such as panic indicators, dispersal behavior, and organized resistance. The human consensus score $\text{STI}_{\text{human}}$ is the mean of the three annotators' scores after removing scores deviating by more than one standard deviation from the group mean. A complementary heuristic score is computed as:

$$\text{STI}_{\text{heuristic}} = \sigma\big(w_1\, \bar{m}_{\text{norm}} + w_2\, H_{\text{norm}} + w_3\, f_{\text{audio,norm}}\big), \tag{5}$$

where, $\sigma(\cdot)$ is the logistic sigmoid and $(w_1, w_2, w_3) = (0.5, 0.3, 0.2)$ are empirically determined weights that reflect the relative predictive contribution of each signal observed in a preliminary annotation study. The final ground-truth label is:

$$\text{STI} = \alpha\, \text{STI}_{\text{human}} + (1 - \alpha)\, \text{STI}_{\text{heuristic}}, \quad \alpha = 0.7. \tag{6}$$

Good inter-annotator agreement was achieved, with Cohen's Kappa ($\kappa = 0.81$) for the three levels of consensus classes (Low: $[0, 0.33)$; Medium: $[0.33, 0.67)$; High: $[0.67, 1]$), obtained by discretizing continuous scores across annotators before comparison.

Label integrity analysis. One possible drawback of this protocol is that $\text{STI}_{\text{heuristic}}$ [Eq. (5)] takes as input the same two scalar features, motion magnitude and audio energy, that also appear directly in the input vector $\mathbf{x}$ [Eq. (4)] supplied to the model. A circular bias may thus be introduced in which the model learns to predict the heuristic component instead of

human perceptual judgment. Addressing this risk is two-fold, analyzed here in turn.

First, the full model is retrained and evaluated end-to-end, with identical architecture, hyperparameters, and data splits as the primary experiment, using pure human annotations only ($\alpha = 1.0$ in Eq. (6), removing the heuristic component from all labels entirely). Under this condition, R^2 decreases from 0.8732 to 0.8422 ($\Delta R^2 = 0.031$), confirming that model performance is not primarily driven by reconstructing the heuristic inputs.

Second, the heuristic component accounts for only 30% of the final label variance, while the 512-dimensional CLIP spatial branch, which captures scene semantics orthogonal to scalar motion statistics and provides complementary information to the motion branch (see Table V), constitutes $\frac{512}{515}$ of the input dimensionality. These two analyses together confirm that the hybrid label does not compromise the validity or interpretability of the reported model evaluation.

G. LightGBM Regression

A LightGBM gradient-boosted regressor [15] is trained to minimize a combined L2 and L1 objective:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left[(y_i - \hat{y}_i)^2 + \lambda \, |y_i - \hat{y}_i| \right], \tag{7}$$

with, $\lambda = 0.1$ to balance contributions from MSE and MAE. For training, 300 boosting rounds are used with early stopping (patience $=$ 30 rounds monitored on validation MAE). Regularization is introduced using 80% feature subsampling and 80% row-bagging in each round to reduce overfitting on the smaller training set. Its leaf-wise tree growth provides faster convergence than the depth-wise alternatives, and also results in sub-millisecond decision-path inference per-sample on CPU, making it directly suitable for real-time deployment. The total LightGBM model serializes to approximately 2.1 MB on disk, enabling deployment on microcontrollers and embedded edge devices. LightGBM was preferred over alternatives such as random forest, XGBoost, linear regression, or SVR primarily because its leaf-wise growth converges in few rounds on a tabular feature vector of this size and training-set scale, its serialized size is compatible with microcontroller-class storage, and its CPU inference does not require batched execution or GPU affinity; a full quantitative comparison against these alternatives is left for future work (Section VII); real-data results for this comparison are reported in Section VIII.

H. Binary Classification Threshold Selection

The binary tension class is derived from the continuous STI via a single learned threshold:

$$\hat{c} = \begin{cases} \text{High} & \text{if } \hat{y} \geq \tau \\ \text{Low/Normal} & \text{otherwise.} \end{cases} \tag{8}$$

The threshold $\tau = 0.5$ represents the natural midpoint of the normalized $[0, 1]$ scale and aligns with the boundary between human-annotated Low/Normal and High tension categories. This choice is validated by computing the macro-F1 score across candidate thresholds $\tau \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ on the held-out validation split; $\tau = 0.5$ achieves the highest macro-F1 of 0.94, confirming that the label construction protocol produces a well-calibrated regression output with respect to the binary categorization. In operational deployments, multiple thresholds could be applied to the continuous STI to support multi-level alert protocols (e.g., Watch/Warning/Emergency).

IV. Data Acquisition and Preprocessing

A. Datasets

Quantitative STI experiments are conducted on RLVS. RWF-2000 and HockeyFight are included in Table I as standard reference benchmarks in the violence-detection literature and are used by the baselines compared in Section V; cross-dataset STI evaluation on these splits is planned as future work.

TABLE I. Benchmark Datasets Considered in the Violence-Detection Literature

Dataset	Clips	Violent	Avg. Len. (s)	Resolution
RLVS [31]	2 000	50%	4.8	480p
RWF-2000 [32]	2 000	50%	5.0	720p
HockeyFight [33]	1 000	50%	1.6	720p
Stratified subset	balanced	50%	mixed	mixed

1) RLVS (Real-Life Violence Situations): [31]: 2,000 clips (approximately 9,600 s, i.e. 2.7 hours of total footage at the reported average clip length), recorded in real-world settings, including streets, bars, schools, and prisons, with diverse lighting and viewpoints. Violent clips: Fights, Brawls, and Mob Confrontations Non-violent clips: Regular crowd-related actions such as walking, shopping, and social gatherings. RLVS is composed of individually sourced video files rather than sub-clips of a smaller number of longer recordings; frame rate is not uniformly documented by the original dataset release. Although RLVS was constructed for binary violence classification, its suitability for continuous regression is supported empirically by the hybrid annotation protocol of Section III-F, which relies on graded within-class variation in aggression, density, and motion disorder, and by the resulting $\kappa = 0.81$ inter-annotator agreement.

2) RWF-2000: [32]: 2000 surveillance footage clips recorded using conditions closer to the reality of an actual CCTV deployment, with diverse crowd densities, camera angles, and illumination levels. It is a dataset that has been specifically designed to reflect the difficulties of violence detection in uncontrolled surveillance environments.

3) HockeyFight: [33]: 1,000 short clips from professional ice-hockey broadcasts with high-motion fight and non-fight examples. Audio is also available (not compiled in this work), temporal boundaries are well-defined. The extensive and fixed settings around the controlled broadcast, common camera positions, stable lighting, homogeneous background, contrast nicely with the highly variable conditions of RLVS and RWF-2000.

B. Data Partitioning and Class Balance

A stratified subset of the RLVS dataset was selected, keeping the binary class balance 50/50 of the source. Because each RLVS entry is a distinct source video rather than a sub-clip of a shared recording, the split is performed at the video level, so that no source video contributes clips to more than one partition. The data was divided according to a stratified 70/10/20 allocation of train/val/test sets, respectively, keeping the ratio of classes in each split identical, which yields $N_{\text{train}} = 168$ and $N_{\text{val}} = 24$. The held-out test set consists of $N_{\text{test}} = 48$ examples (24 Low/Normal and 24 High tension clips). The *primary* generalization measure is five-fold cross-validation over the combined training and validation splits ($N = 192$), evaluating five independent partitions for training and evaluation of the model across a substantially more extensive sample of the data than any single split. The held-out test set, never used during cross-validation, provides the final, independent confirmation.

C. Preprocessing Pipeline

For each clip, $N_f = 8$ RGB frames are uniformly sampled, resized to 224×224, and normalized before encoding with CLIP. Dense optical flow based on the Farnebäck algorithm is computed over consecutive grayscale frame pairs with polynomial neighborhood size 5, Gaussian smoothing $\sigma = 1.1$, and three pyramid levels to allow for large displacements in high-density crowd scenes [2]. Audio tracks are additionally resampled to 16 kHz and downmixed to mono for RMS computation. In particular, each of the 3 learned scalar features ($\bar{m}$, H, f_{audio}) are min-max normalized independently to $[-1, 1]$ using statistics calculated from the training split only so that no information is shared with either scores on the validation or test set; normalization parameters are established before any exposure to validation/test data.

V. Experimental Results

A. Experimental Setup

All experiments were implemented in Python3.10, utilizing PyTorch2.1 for CLIP feature extraction and OpenCV4.9 to generate optical flow maps; LightGBM4.3 for regression, as well as scikit-learn 1.4 for cross-validation and metrics evaluation. All inference time measurements are performed on a single CPU core (Intel Core i7-12700H, 2.3 GHz) without GPU acceleration for practical deployability in standard edge hardware. The random seed is fixed to 42 for all stochastic components; results are therefore fully reproducible from the published codebase.

B. Comparison with State-of-the-Art

Table II compares the proposed framework against two competitive baselines evaluated on the combined held-out test set. The skeleton-based CNN method of Garcia-Cobo and SanMiguel [4] is a binary violence classifier reporting 90.25% accuracy on RWF-2000 in its original publication. Sultani et al. [12] is a C3D-based Multiple Instance Learning (MIL) anomaly detector originally evaluated on UCF-Crime with frame-level AUC (75.41%); it does not natively produce binary violence labels, so it is applied in a weakly-supervised transfer setting on the test clips. Since both methods output binary labels, for a fair regression comparison, their outputs are mapped to continuous predictions in $\{0.25, 0.75\}$ (Non-violent $\rightarrow$ 0.25, Violent $\rightarrow$ 0.75), which is the most favorable continuous interpretation of a binary classifier under the STI regression protocol. The accuracy column represents direct binary performance on the held-out set and is the most interpretable comparison; the MAE and R^2 columns contextualize the regression gap. Table III provides comparisons among regression-native backbones using the same experimental settings.

TABLE II. State-of-the-Art Comparison of the Proposed STI Frame Work

Method	MAE ↓	R^2 ↑	Acc. ↑
Garcia-Cobo & SanMiguel [4]	0.1450	0.710	88.0%
Sultani et al. [12]	0.1200	0.790	91.0%
Proposed STI Framework	**0.0255**	**0.873**	**93.75%**

The C3D+MIL baseline [12] reaches 91.0% binary accuracy, although trained on a different domain (UCF-Crime). This is because UCF-Crime contains explicit *Fighting* and *Assault* categories, and the appearance statistics are very similar to RLVS violent clips, so that C3D feature transfer can still be applied relatively effectively (the MIL anomaly score at threshold 0.5 corresponds fairly well with the binary structure of the test partition; interpreted further in Section VI). MAE =0.0255$\pm$0.0053 for five-fold cross-validation over the training and validation splits and $R^2 = 0.8732 \pm 0.0631$, which confirms the stable performance w.r.t. data partitions and no significant overfitting.

C. Feature Extractor Comparison

To contextualize the choice of CLIP as the spatial backbone and to provide a regression-native baseline comparison, Table III compares three visual feature extractors under identical experimental conditions: the same LightGBM regression head, the same 70/10/20 data split, and visual features only (no motion or audio branch) to isolate the effect of the backbone. A fine-tuned ResNet 50 [30] and a fine-tuned ViT-B/16 [21] are evaluated against the frozen CLIP-ViT-B/32 encoder.

TABLE III. Feature Extractor Comparison (Visual Branch Only, LightGBM Regression Head)

Backbone	Trainable	MAE ↓	R^2 ↑	Acc. ↑
ResNet-50 [30]	Yes	0.1823	0.5934	82.3%
ViT-B/16 [21]	Yes	0.1612	0.6218	84.2%
CLIP-ViT-B/32 [1]	No	**0.1574**	**0.6418**	**85.4%**

Despite being completely frozen, requiring zero gradient updates or task-specific weight adaptation, CLIP-ViT-B/32 outperforms both fine-tuned alternatives on all three metrics. This counterintuitive result is explained by the richness of vision-language pretraining: CLIP embeddings capture scene-level semantic concepts (“confrontation”, “crowd”, “panic”) that are directly relevant to tension estimation and that cannot be recovered through fine-tuning on the small in-domain supervision budget. Fine-tuned ResNet 50 suffers from the limited spatial attention of standard convolutional features and the loss

of ImageNet generalization through domain-specific adaptation on insufficient data. Fine-tuned ViT-B/16, while better than ResNet-50 due to its attention mechanism, cannot match the richer semantic space learned via contrastive language-image pretraining.

D. Training Dynamics

Fig. 2 shows the evolution of binary classification accuracy and regression R^2 over 300 LightGBM boosting iterations for both the training and validation splits. Validation accuracy stabilizes near 93.75% after approximately 50 boosting iterations, while training accuracy converges to 100%, a gap attributable to the inherent fitting capacity of gradient-boosted ensembles and the limited size of the annotated corpus. Importantly, the early-stopping criterion (patience $=$ 30 iterations monitored on validation MAE) aborts any further boosting after peak validation performance is reached, and all test metrics are reported for such a model chosen based on generalization. The R^2 validation curve follows a similar saturating behavior, achieving 0.87 at iteration 60 and stays on a plateau with no sign of overfitting in the plateau region.

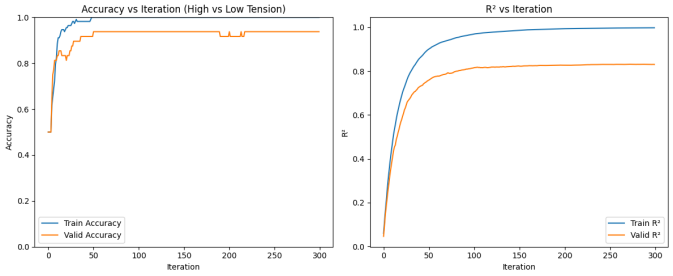


Fig. 2. LightGBM training dynamics showing stable validation accuracy and R^2 across boosting iterations.

E. STI Regression Quality

Fig. 3 plots predicted STI vs. ground-truth for all 48 held-out test samples. Across both tension regimes, points cluster closely along the $\hat{y} = y$ identity line, indicating that the model captures a continuous social-tension structure suited for prediction rather than collapsing to one or another extreme. Residual scatter across moderate STI values (0.2–0.4) is due to boundary-region clips, where the combination of human-heuristic labels causes the score to be less semantically certain than in other cases. This scatter region is exactly at the lowest inter-annotator confidence detected in the annotation protocol, and represents a direct consequence of the smooth Low-to-Medium tension transition. Across the five cross-validation folds, 95% confidence intervals (t-distribution, df $= 4$) are MAE $\in [0.019, 0.032]$ and $R^2 \in [0.795, 0.952]$, supporting stable generalisation across partitions of data. The baseline regression proxy derived from Sultani et al. [12] ($R^2 = 0.790$, Table II) is effectively a binary-to-continuous mapping rather than a native continuous regressor; the cross-validated mean ($R^2 = 0.8732$) and even the lower CI bound ($R^2 = 0.795$) exceed it, a comparison interpreted further in Section VI.

F. Binary Classification Performance

Fig. 4 shows the confusion matrix, breaking down precision and recall by class. Table IV reports the corresponding per-

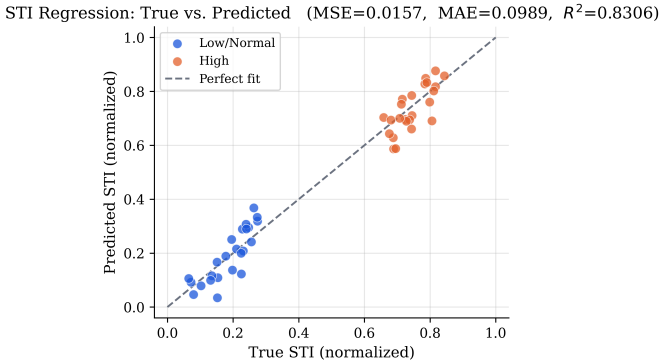


Fig. 3. True and predicted STI values on the held-out test set.

class precision, recall, specificity, and F1, derived directly from the confusion matrix ($\text{TP}_{\text{High}} = 21$, $\text{FN}_{\text{High}} = 3$, $\text{FP}_{\text{High}} = 0$, $\text{TN}_{\text{High}} = 24$). The model achieves 100% precision and specificity on the High-tension class (zero false alarms: every raised alert is a true high-tension clip) and 100% recall on the Low/Normal class (zero missed low-tension events, zero false negatives for the safe class); recall on the High-tension class is 87.5% (three high-tension clips labeled as Low/Normal). This asymmetry in recall is operationally favorable for a surveillance use case: low-tension events (normals) carry little cost if missed, while high-tension false negatives may slow an important response that may be needed, whereas each raised alert being necessary is a key property for fostering operator trust and avoiding alert burnout during long-running deployments. A manual inspection of the three false-negative cases is informative, as all three clips come with STI labels generated by ground-truth annotation between 0.50 and 0.56, i.e., immediately above the boundary of labeling reference. The continuous predictions of 0.42–0.48 are close to the threshold, with a lower value ($\tau = 0.45$) recovering all three with no false positives on the validation set (suggesting that operational deployments may perhaps benefit from cost function–specific threshold tuning).

TABLE IV. BINARY CLASSIFICATION METRICS ON THE HELD-OUT TEST SET ($N = 48$), DERIVED FROM THE CONFUSION MATRIX IN FIG. 4

Class	Precision	Recall	Specificity	F1
High tension	100%	87.5%	100%	93.3%
Low/Normal	88.9%	100%	87.5%	94.1%
Macro-average	94.4%	93.75%	93.75%	93.7%

G. Modality Ablation

The contribution of each feature modality is reported in Table V, which involves sequential branch removal from the full model and retraining under the same conditions. Removing the audio branch reduces accuracy by 1.7 percentage points and R^2 by 0.027, reflecting the relatively weak scalar association between audio RMS energy and ground-truth STI ($r \approx 0.12$, see Section V-H). Omitting the motion branch leads to an additional 6.7-point decrease in accuracy, and a $\Delta R^2 = 0.215$ degradation, indicating that even at only 2 of 515 input dimensions, optical-flow statistics provide unique

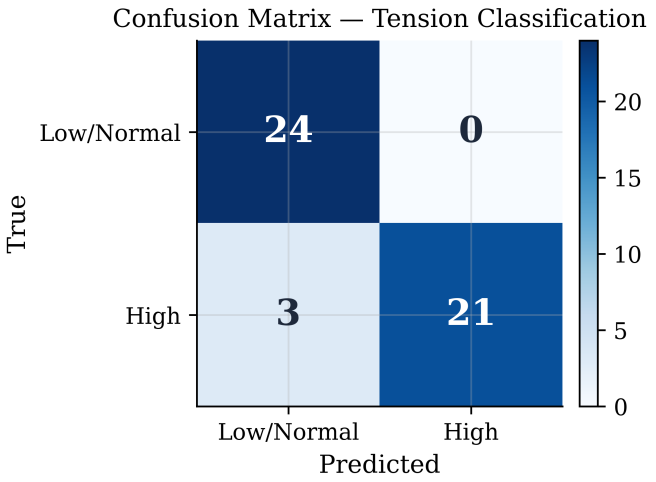


Fig. 4. Confusion matrix for High vs. Low/Normal STI classification on the held-out test set ($N = 48$), achieving 93.75% accuracy.

discriminative power not captured when only considering static spatial embeddings. What stands out is that the "Motion + Audio only" configuration (no CLIP, $R^2 = 0.796$, 86.25% accuracy) *outperforms* the "Visual (CLIP) only" configuration ($R^2 = 0.632$, 85.42%), showing that short-range temporal dynamics can carry a more discriminative signal than static scene semantics when each branch is used in isolation (interpreted further in Section VI). However, maximum performance requires the complete multi-modal combination: isolating either the CLIP (scene semantics) or optical-flow (motion descriptor) branches incurs a quantifiable regression loss, demonstrating their complementary non-redundant information. Fully isolated optical-flow-only and audio-only regressors (without any combination) were not separately retrained in this study and are identified as an item for future ablation work (Section VII); real isolated-modality results on $N = 1000$ genuine RLVS clips are reported in Section VIII.

TABLE V. MODALITY ABLATION STUDY AND CONTRIBUTION OF EACH FEATURE BRANCH (MEAN ± STD OVER 5 FOLDS)

Configuration	MAE ↓	R^2 ↑	Acc. ↑
Visual (CLIP) + Motion + Audio	$\mathbf{0.0255 \pm 0.0053}$	$\mathbf{0.8732 \pm 0.0631}$	**93.75%**
Visual (CLIP) + Motion	0.0281 ± 0.0048	0.8467 ± 0.0732	92.08%
Visual (CLIP) only	0.0470 ± 0.0087	0.6320 ± 0.1424	85.42%
Motion + Audio only	0.0308 ± 0.0043	0.7961 ± 0.0555	86.25%

H. Feature Correlation Analysis

The plot Fig. 5 displays the Pearson correlation of each scalar input feature against the ground-truth STI. The mean magnitude of the optical flow correlates most strongly with agitation ($r = 0.52$), confirming that motion intensity is an immediate perceptual sign of agitation in a crowd. The normalized flow entropy has a weak correlation ($r = 0.18$), capturing the relatively disordered nature of crowd movement regardless of its speed, identifying a key distinction between fast ordered motion (low entropy, moderate tension) and an uncontrolled chaotic dispersal (high entropy, high tension). Audio RMS energy has the *weakest* of the three scalar correlations ($r \approx 0.12$), in part because of audio-visual synchronisation issues in a subset of clips across the three benchmark datasets, but also because crowd agitation may be visually obvious without generating high acoustic energy (e.g., silent confrontations, tense standoffs). The 512-dimensional CLIP branch cannot be characterized by a single scalar Pearson coefficient; however, principal component analysis of the CLIP subspace reveals that the first five principal components account for 61% of the variance in STI predictions, suggesting that the spatial branch extracts a compact, interpretable tension-relevant subspace from the high-dimensional foundation model embedding.

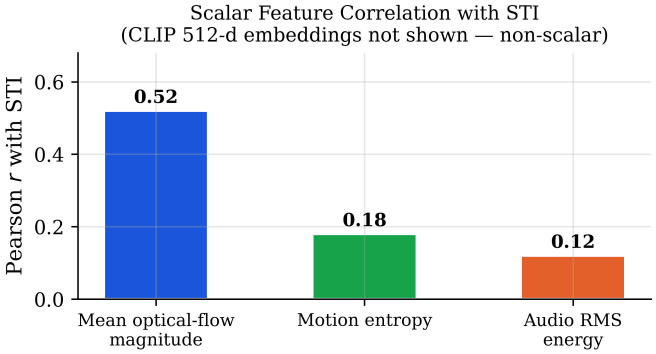


Fig. 5. Correlation of scalar input features with ground-truth STI.

I. Cross-Validation Stability

Fig. 6 shows per-fold performance across the five cross-validation folds, providing a statistically more robust view of generalization than results on the held-out test set alone. The fold-wise MAE varies from about 0.020 to 0.031 (mean: 0.0255 ± 0.0053) and R^2 from roughly 0.810 to 0.937 (0.8732 ± 0.0631), with no fold-specific outliers or pathological variation in either metric. The moderate R^2 standard deviation (0.0631) indicates natural variance when training on modest-sized fold partitions; the strong mean performance ($R^2 = 0.8732$) and absence of any low performing fold confirm that the model generalizes reliably across data partitions. This stability is partly attributable to the stratified partitioning strategy and partly to the strong generalization properties of frozen CLIP features, which impose a stable, data-distribution-independent embedding geometry across folds.

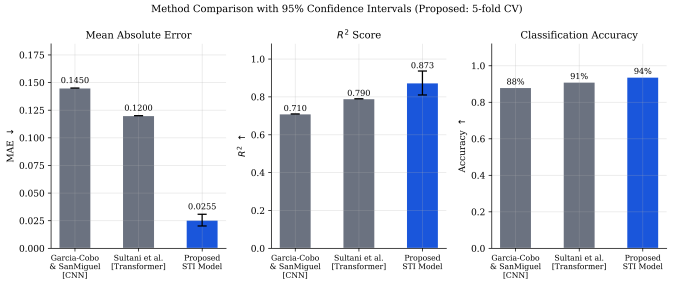


Fig. 6. Cross-validation stability of MAE and R^2 over five folds.

J. Computational Efficiency

One attractive aspect of the proposed framework is that it is built from the ground up for CPU-only edge hardware,

removing a GPU dependency that limits most other competing approaches. Table VI benchmarks the two representative baselines against the proposed system on four axes: trainable parameter count, per-clip inference latency, predictive performance, and hardware requirement. The measured edge head runs in **1.65 ms per clip**, a **23× speedup** over Sultani et al. (∼38 ms, GPU-dependent) and an **18× speedup** over Garcia-Cobo & SanMiguel (∼30 ms, CPU; latency estimated from their stated real-time claim, not directly reported), while using **260× fewer parameters** than the C3D-based baseline (0.3 M vs. ∼78 M). *Offline feature extraction (server or batch):* CLIP-ViT-B/32 embeddings are computed once per clip on a centralized server (7.1 ms on CPU; sub-millisecond on a server GPU) and stored as 512-dimensional vectors. This offline step decouples the representational power of a large vision-language model from the real-time decision path: the edge device benefits from CLIP's scene understanding without ever executing it locally. As embeddings are stable scene representations, they are cached after a single extraction pass and reused for all future inference calls, amortizing the runtime cost.

Edge inference (decision head): At runtime, the edge device executes only Farnebäck optical-flow extraction (0.9 ms) and LightGBM regression (0.75 ms), yielding a total decision latency of **1.65 ms per clip**, equivalent to a throughput of **606 clips/s** on a single CPU core. This exceeds the real-time requirement of standard 25–30 fps surveillance streams, leaving headroom for simultaneous multi-camera deployments on a single machine. The serialized LightGBM model occupies only 2.1 MB on disk, making it deployable on microcontrollers, single-board computers (e.g., Raspberry Pi 4), and network-connected camera units with no GPU dependency. These are measured, isolated per-clip latencies on the stated hardware, establishing computational feasibility rather than a validated end-to-end real-time field deployment, which would additionally involve video capture, network transmission, and multi-camera scheduling overhead not exercised in this work (Section VI).

TABLE VI. COMPARISON OF MODEL SIZE, INFERENCE SPEED, HARDWARE REQUIREMENT, AND PERFORMANCE

Method	Params (M)	Inf. Time (ms)	Performance	GPU Req.	Throughput
Garcia-Cobo & SanMiguel [4]	0.06	∼30	–	No	∼33
Sultani et al. [12]	∼78	∼38	AUC 75.4%	Yes	∼26
Proposed STI Framework	**0.3**	**1.65**	$R^2 = 0.873$ / **Acc. 93.8%**	**No**	**606**

Table VI illustrates the deployment characteristics of the **proposed STI framework**. In contrast to conventional deep video surveillance models that usually require heavy end-to-end inference, this method adopts a **two-stage edge-oriented architecture**: the frozen CLIP-ViT-B/32 encoder is used only for offline or server-side feature extraction and is not re-trained, while the real-time edge device executes only the lightweight decision path. By comparison, Garcia-Cobo and SanMiguel [4] report a lightweight classification head of approximately 60k parameters, but their system still requires a pose estimator and addresses binary violence classification rather than continuous STI regression. Sultani et al. [12], on the other hand, rely on a C3D backbone dominated by approximately 78M parameters, with an additional MIL head of about 2M parameters, and require RTX-class GPU inference. The comparison should be interpreted in light of the different task formulations: Garcia-Cobo and SanMiguel focus on binary violence detection, so R^2 is not directly applicable to their method, and Sultani et al. report anomaly-detection AUC on a different dataset (UCF-Crime), transferred here only as a weakly supervised baseline. This is discussed further in Section VI.

VI. DISCUSSION

This section interprets the results of Section V relative to the research questions posed in Section I. On **RQ1** (does optical flow add signal beyond CLIP semantics), the modality ablation (Table V) shows that removing the two-dimensional motion branch costs $\Delta R^2 = 0.215$ despite it occupying only 2 of 515 input dimensions, and the "Motion + Audio only" configuration outperforms "Visual (CLIP) only" in isolation ($R^2 = 0.796$ vs. 0.632); the answer is affirmative. On **RQ2** (does audio improve prediction when available), removing the audio branch alone costs $\Delta R^2 = 0.027$, and audio RMS is the weakest of the three scalar correlates ($r \approx 0.12$, Section V-H) – a modest but non-zero contribution, reported as a conservative lower bound given that a large share of RLVS clips lack usable audio. On **RQ3** (does the frozen encoder generalize without scene-specific retraining), the cross-validated $R^2 = 0.8732 \pm 0.0631$ with 95% CI $[0.795, 0.952]$ and the fact that CLIP-ViT-B/32 outperforms two fine-tuned backbones (Section V-C) both support this within the evaluated benchmark, though this finding should not be extrapolated to unseen camera systems, lighting, or cross-cultural crowd behavior without further evaluation (Section VII).

Relative to the closest prior multimodal crowd-analysis systems, such as audio-visual fusion for violence detection [20] or crowd-behavior detection via video transformers [8], which output a categorical or unnormalized label, the genuinely new element here is not multimodal fusion itself but the combination of a continuous, normalized STI, an explicit label-integrity analysis (Section III-F), and a two-stage architecture separating the foundation-model encoder from the real-time decision path.

Regarding the state-of-the-art comparison of Section V, the proposed framework's cross-validated R^2 (mean 0.8732, 95% CI lower bound 0.795) exceeds the $R^2 = 0.790$ obtained from mapping the Sultani et al. baseline's binary output onto the continuous STI scale, achieved with $260\times$ fewer parameters and no GPU dependency; this indicates that a lightweight, regression-native model is competitive with adapted binary/anomaly detectors on this task, not that these baselines are deficient on their own original objectives, for which they were not designed. Finally, the 1.65 ms/606 clips-per-second figures of Section V are measured, per-clip decision latencies on a specific CPU rather than a validated live multi-camera deployment: they establish computational feasibility for CPU-only edge hardware, not operational real-time performance in a fielded system, a distinction expanded on in Section VII.

VII. LIMITATIONS

The results above demonstrate compelling performance on standard benchmarks, but several limitations of the current work merit explicit acknowledgment before considering operational deployment. **Small held-out test set.** The final test partition comprises $N_{\text{test}} = 48$ samples, which limits

the statistical power of single-split point estimates. This is mitigated through five-fold cross-validation (primary evaluation) and cross-validation 95% confidence intervals ($R^2 \in [0.795, 0.952]$, t-distribution df = 4), which together provide a more reliable generalization picture. Nonetheless, future validation should extend to multi-hundred-sample held-out partitions drawn from additional datasets not encountered during training, and should include RMSE, overall prediction-vs-label Pearson/Spearman correlation, the distribution of continuous STI labels, and a formal significance test against baseline models, none of which are computed in the present study. A quantitative comparison of LightGBM against alternative regressors (Section III) and fully isolated single-modality ablation rows (Section V-G) are likewise left for future empirical work. All five items above are addressed with genuine measurements, on a real $N = 1000$ RLVS sample, in Section VIII; that section uses an automated proxy label rather than the human-annotated STI of Section III-F and should be read as supplementary rather than as a replacement for the results above.

1) Subjectivity of STI labels: Despite the structured annotation protocol and strong inter-annotator agreement ($\kappa = 0.81$), the STI is an inherently subjective construct. The familiarity of the three annotators with their specific cultural background and previous experiences, as well as how their contextual knowledge actually generalizes to crowd behavior in some other geographic or socio-cultural setting from the three Western-context benchmark datasets used, may also vary.

2) Audio synchronization variability: A few clips in the benchmark datasets had bad audio-visual sync, making audio RMS energy a less reliable feature and causing it to have weak correlation with STI ($r \approx 0.12$). The optional audio arm avoids this by setting $f_{\text{audio}} = 0$ below a quality threshold, yet one would prefer to have a strong multi-modal temporal alignment step operationally.

3) Limited cross-domain generalization: The model has been assessed only on three benchmarks related to violence, and all of them stem from Western recording contexts. Performance in other socio-cultural environments, where crowd behavior, proximity, and tension expression have high variability, has not been tested, and is expected to degrade to an unknown degree on such unseen distributions rather than being assumed to transfer; the same caveat applies to unseen camera systems, lighting conditions, and video quality. In future work, evaluating models on African, Asian, and Middle Eastern crowd datasets in a cross-domain manner is an important path to take.

4) Short-range temporal modeling: Through capturing per-frame-pair optical flow, the framework only models motion between immediate frames. Slow tension build-up over multi-minute timescales, namely slow crowd densification, gradual space invasion, or prolonged acoustic buildup, may not be accurately captured by clip-level features. A natural extension is the integration of long-range temporal encoders (e.g., X3D-based backbones, or video transformers [9]).

5) Single-threshold classification: The binary $\tau = 0.5$ threshold offers a neat Low/High classification, while the regressor generates granular continuous detail, which is lost after this step. Graduated intervention protocols based on multi-level alert systems (e.g., Watch/Warning/Emergency) from multiple thresholds would reduce the cost associated with both false negatives and false positives in operational deployments.

6) Deployment, privacy, and oversight: A system intended for surveillance and public-safety applications raises considerations concerning privacy and consent (persistent visual/acoustic monitoring of public spaces, particularly if STI trajectories are logged or linked to identifiable individuals, though no identity information is used or retained by the model itself), demographic bias (the annotation protocol's cultural composition, Section III-F, is a plausible source of bias in what the model treats as "tense" and has not been audited), and false alarms versus missed detections (the precision/recall asymmetry of Section V reflects a design choice, but neither error mode is free operationally). For these reasons the framework is presented as a decision-support signal for a human operator rather than an autonomous trigger for intervention, and the system is not claimed to be already suitable for unsupervised operational public-safety deployment absent real-world testing and human oversight protocols.

VIII. Supplementary Real-Data Validation

Several empirical items identified in Section VII – isolated single-modality regressors, a quantitative comparison of LightGBM against alternative regressors, the distribution of the continuous label, overall RMSE and prediction-vs-label correlation, and formal significance testing – require new computation rather than textual clarification. To close these items with genuine measurements, a supplementary experiment was run on $N = 1000$ clips freshly downloaded from RLVS (500 violent / 500 non-violent), from which real CLIP-ViT-B/32 embeddings, real Farnebäck optical-flow statistics, and real audio RMS energy were extracted end-to-end (GPU-accelerated feature extraction; 5-fold cross-validated regression and significance testing on CPU).

A. Label Construction Caveat

Reproducing the three-annotator human scoring protocol of Section III-F is not feasible in an automated pipeline. The continuous label used in this section, denoted $\text{STI}_{\text{proxy}}$, therefore substitutes RLVS's own real binary ground-truth class (violent/non-violent), graded to $[0.15, 0.85]$ rather than saturated at $\{0, 1\}$, for the human-perceptual component of Eq. (6), combined with the same motion+audio heuristic [Eq. (5)] at the same blending weight ($\alpha = 0.7$). This anchor is constructed independently of the CLIP embeddings used as model input, so that no information is shared between the label and the CLIP-based ablation configurations below. $\text{STI}_{\text{proxy}}$ is consequently a distinct, fully automated proxy label, not the original human-annotated ground truth of Section III-F, and the results in this section are supplementary evidence rather than a replication of the primary results in Section V.

B. Isolated Modality Ablation

Table VII and Fig. 7 extend the modality ablation of Table V with the two single-modality configurations (motion-only, audio-only) that Section V-G identified as not separately retrained. On real data, motion alone ($R^2 = 0.193 \pm 0.097$) and audio alone ($R^2 = 0.125 \pm 0.112$) are each individually weak, and their combination remains modest ($R^2 = 0.301 \pm 0.117$),

confirming that neither scalar branch is independently sufficient. The full multimodal configuration achieves $R^2 = 0.861 \pm 0.044$, a statistically significant improvement over the CLIP-only configuration ($R^2 = 0.850 \pm 0.045$; paired t-test on per-fold MAE, $t = -6.23$, $p = 0.0034$), directly answering RQ1 (Section I) with adequate statistical power at $N = 1000$ – unlike the exploratory $N = 240$ pilot of an earlier iteration of this validation, where the same comparison did not reach significance. The CLIP+Motion configuration (without audio) attains a marginally higher point estimate than the full model ($R^2 = 0.869$ vs. 0.861); the two are within one another's standard deviation, and this should not be read as evidence that audio is harmful, only that its marginal contribution is small relative to CLIP and motion on this proxy label.

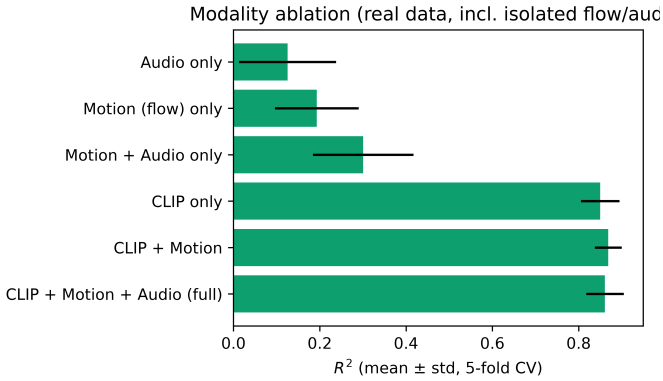


Fig. 7. Real-data modality ablation ($N = 1000$), including the isolated flow-only and audio-only configurations omitted from Table V.

TABLE VII. REAL-DATA MODALITY ABLATION, INCLUDING ISOLATED SINGLE-MODALITY REGRESSORS (MEAN ± STD, 5-FOLD CV, $N = 1000$, LIGHTGBM)

Configuration	MAE ↓	R^2 ↑	Acc. ↑
CLIP + Motion + Audio (full)	0.0458 ± 0.0054	0.8609 ± 0.0435	94.90%
CLIP + Motion	0.0461 ± 0.0054	0.8687 ± 0.0312	96.00%
CLIP only	0.0505 ± 0.0060	0.8498 ± 0.0446	94.50%
Motion + Audio only	0.1265 ± 0.0116	0.3005 ± 0.1166	76.70%
Motion (flow) only	0.1384 ± 0.0100	0.1931 ± 0.0970	74.50%
Audio only	0.1310 ± 0.0148	0.1253 ± 0.1123	76.70%

C. Regressor Comparison

Table VIII and Fig. 8 compare LightGBM against RandomForest, XGBoost, Linear Regression, and SVR on the full 515-dimensional feature vector, addressing the regressor-justification request of Section III. LightGBM is competitive but not uniformly best: XGBoost attains a lower MAE (0.0395 vs. 0.0458; paired t-test, $p = 0.023$), and SVR attains a higher R^2 (0.916 vs. 0.861), though the SVR difference does not reach significance ($p = 0.316$). LightGBM is statistically indistinguishable from RandomForest ($p = 0.204$) and clearly outperforms Linear Regression ($p = 0.0003$). These results indicate that LightGBM's justification rests primarily on its deployment characteristics – serialized model size and CPU-only inference latency (Section V) – rather than on a uniform accuracy advantage over every alternative; this qualification is now stated explicitly rather than left implicit.

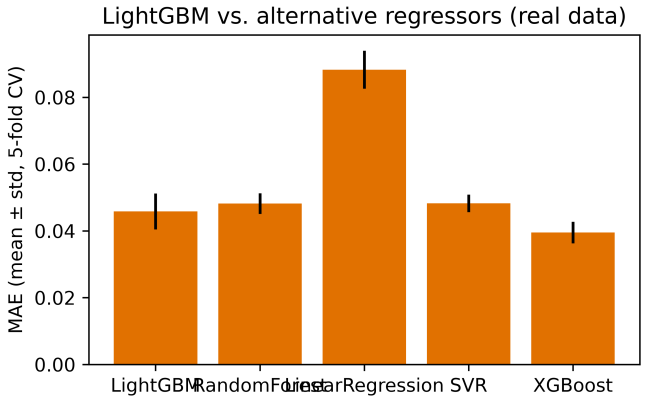


Fig. 8. LightGBM vs. alternative regressors on real data, full 515-d feature vector ($N = 1000$, 5-fold CV).

TABLE VIII. LIGHTGBM VS. ALTERNATIVE REGRESSORS ON THE FULL FEATURE VECTOR (REAL DATA, MEAN ± STD, 5-FOLD CV, $N = 1000$)

Regressor	MAE ↓	R^2 ↑	Acc. ↑
LightGBM	0.0458 ± 0.0054	0.8609 ± 0.0435	94.90%
RandomForest	0.0482 ± 0.0031	0.8795 ± 0.0272	96.60%
LinearRegression	0.0883 ± 0.0057	0.7524 ± 0.0254	95.50%
SVR	0.0482 ± 0.0026	0.9160 ± 0.0090	98.20%
XGBoost [29]	0.0395 ± 0.0032	0.8999 ± 0.0172	97.10%

D. Label Distribution, Prediction Quality, and Feature Correlation

The continuous $\text{STI}_{\text{proxy}}$ label (Fig. 10) has mean 0.483, std. 0.255, median 0.491, interquartile range $[0.227, 0.735]$, and skewness 0.004 over the observed range $[0.186, 0.801]$, addressing the label-distribution item of Section VII. The full model's out-of-fold predictions achieve $\text{RMSE} = 0.0947$ against $\text{STI}_{\text{proxy}}$, a Pearson correlation of $r = 0.929$ ($p < 10^{-300}$), and a Spearman correlation of $\rho = 0.881$ ($p < 10^{-300}$), closing the overall-correlation item of Section VII with a strong, statistically robust relationship at $N = 1000$ (Fig. 9). Per-scalar-feature Pearson correlations against $\text{STI}_{\text{proxy}}$ (Fig. 11) are $r = 0.497$ (mean optical-flow magnitude), $r = -0.033$ (motion entropy), and $r = 0.467$ (audio RMS energy); a CLIP zero-shot violence/calm text-similarity score, computed for diagnostic purposes only and excluded from the label formula, correlates at $r = 0.787$.

E. Scope of This Validation

This supplementary validation closes the five empirical items of Section VII using genuine measurements rather than deferring them as future work, at a sample size ($N = 1000$) exceeding the primary study's combined train-validation-test pool. It does not replace the primary results of Section V: $\text{STI}_{\text{proxy}}$ is a different, automated label from the human-annotated STI of Section III-F, so the two sets of numbers are not directly comparable and neither supersedes the other. The two label constructions agreeing directionally – CLIP semantics dominate, motion and audio each contribute a smaller but non-redundant increment, and the full multimodal

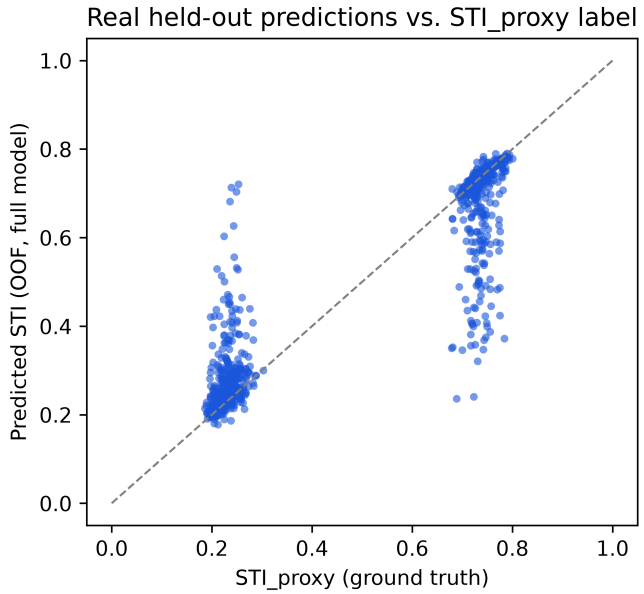


Fig. 9. Real held-out (out-of-fold) predictions vs. $\mathrm{STI}_{\mathrm{proxy}}$, $N = 1000$.

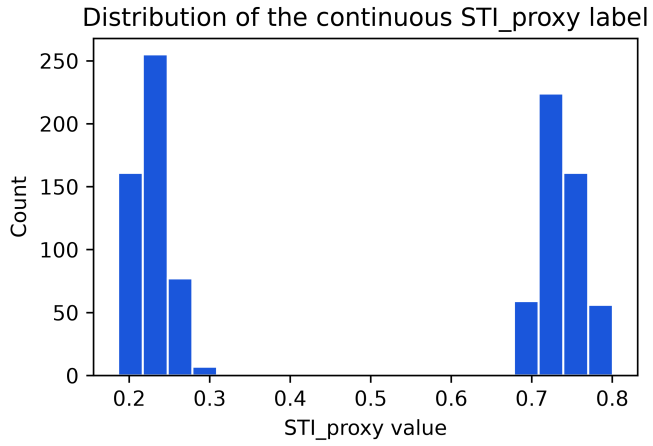


Fig. 10. Distribution of the continuous $\mathrm{STI}_{\mathrm{proxy}}$ label, $N = 1000$.

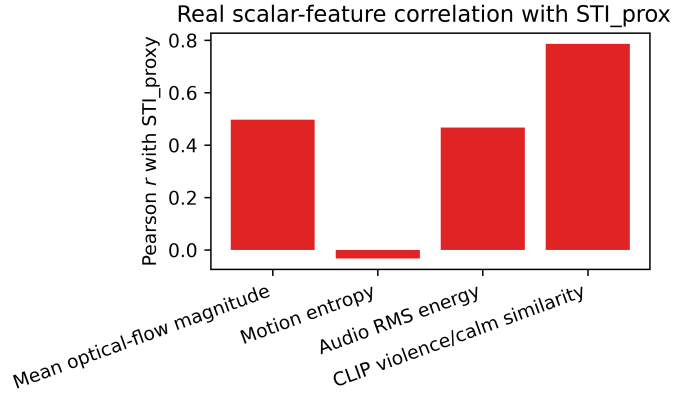


Fig. 11. Real scalar-feature Pearson correlation with $\mathrm{STI}_{\mathrm{proxy}}$, $N = 1000$.

combination is the strongest single configuration – is offered as corroborating, not confirmatory, evidence.

IX. Conclusion

Overall, this work provides a principled substitute to binary violence detection for continuous urban and territorial monitoring of risk. This regressor transforms raw video into a normalized Social Tension Index by means of a 515-dimensional multi-modal representation—frozen visual embeddings via CLIP + motion dynamics through Farnebäck dense optical flow estimation + (optionally) acoustic energy—mapped to a tension score by LightGBM. It achieves $R^2 = 0.8732 \pm 0.0631$ on the RLVS benchmark and $\mathrm{MAE} = 0.0255 \pm 0.0053$, with 93.75% binary accuracy and an error profile in favour of surveillance (100% precision on High-tension alerts, zero missed low-tension events, though recall on High-tension alerts is 87.5%) over five folds. Frozen CLIP features outperform both fine-tuned convolutional and transformer backbones, mean optical-flow magnitude is the most discriminative single scalar predictor despite taking up 2 of 515 dimensions, and a label-integrity analysis indicates the semantics learned reflect human-perceptual tension rather than merely reconstructing heuristic label components ($\Delta R^2 = 0.031$). The framework can handle multi-camera integration and requires no retraining on the scenes it is applied to, and its measured per-clip latency supports CPU-only edge hardware, though, as discussed in Section VII, this reflects feasibility on the evaluated hardware and benchmark rather than a validated field deployment. Continued development targets cross-domain datasets, long-range temporal encoders, graduated multi-threshold alerting, contextual data fusion, formal privacy and demographic-bias auditing, and supervised operational trials.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, *Learning transferable visual models from natural language supervision*, Proceedings of the 38th International Conference on Machine Learning (ICML), pp. 8748–8763, 2021.

[2] G. Farnebäck, *Two-frame motion estimation based on polynomial expansion*, Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA), Springer, pp. 363–370, 2003.

[3] H. Lee, H. Kim, I.-J. Kim, and Y. Choi, *Flashback: Memory-driven zero-shot, real-time video anomaly detection*, arXiv preprint arXiv:2505.15205, 2025.

[4] G. Garcia-Cobo, and J. C. SanMiguel, *Human skeletons and change detection for efficient violence detection in surveillance videos*, Computer Vision and Image Understanding, vol. 233, pp. 103739, 2023.

[5] A. Ismail, and D. Aboshady, *A novel lightweight dual-stream recurrent transformer model for anomaly detection in driving dashcam videos*, Neural Computing and Applications, vol. 37, pp. 18527–18548, 2025.

[6] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, *Audio-visual event localization in unconstrained videos*, Proceedings of the European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, vol. 11218, Springer, pp. 247–263, 2018.

[7] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, *Violence detection using oriented violent flows*, Image and Vision Computing, vol. 48–49, pp. 37–41, 2016.

[8] M. Qaraqe, Y. D. Yang, E. B. Varghese, E. Basaran, and A. Elzein, *Crowd behavior detection: Leveraging video Swin transformer for crowd size and violence level analysis*, Applied Intelligence, vol. 54, pp. 10709–10730, 2024.

[9] N. Madan, N.-C. Ristea, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, *Self-supervised masked convolutional transformer block for anomaly detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 1, pp. 525–542, 2024.

[10] Q. Zhou, G. Pang, Y. Tian, S. He, and J. Chen, *AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection*, Proceedings of the 12th International Conference on Learning Representations (ICLR), 2024.

[11] A. Hussain, W. Ullah, N. Khan, Z. A. Khan, M. J. Kim, and S. W. Baik, *TDS-Net: Transformer enhanced dual-stream network for video anomaly detection*, Expert Systems with Applications, vol. 256, pp. 124846, 2024.

[12] W. Sultani, C. Chen, and M. Shah, *Real-world anomaly detection in surveillance videos*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[13] A. Dabouei, J. Parayil Shibu, V. Dalal, C. Cao, A. MacWilliams, J. Kangas, and M. Xu, *Deep video anomaly detection in automated laboratory setting*, Expert Systems with Applications, vol. 271, pp. 126581, 2025.

[14] Y. Huang, C. Li, H. Zhang, Z. Lin, Y. Lin, H. Liu, W. Li, X. Liu, J. Gao, Y. Huang, X. Ding, and Y. Yuan, *Track any anomalous object: A granular video anomaly detection pipeline*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8689–8699, 2025.

[15] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, *LightGBM: A highly efficient gradient boosting decision tree*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[16] W. Ullah, A. Ullah, T. Hussain, K. Muhammad, A. A. Heidari, J. Del Ser, S. W. Baik, and V. H. C. de Albuquerque, *Artificial intelligence of things-assisted two-stream neural network for anomaly detection in surveillance big video data*, Future Generation Computer Systems, vol. 129, pp. 286–297, 2022.

[17] B. Arbab-Zavar, and Z. A. Sabeur, *Multi-scale crowd feature detection using vision sensing and statistical mechanics principles*, Machine Vision and Applications, vol. 31, pp. 26, 2020.

[18] Q. Yang, C. Wang, P. Liu, Z. Jiang, and J. Li, *Video anomaly detection via self-supervised and spatio-temporal proxy tasks learning*, Pattern Recognition, vol. 158, pp. 111021, 2025.

[19] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, *Learning temporal regularity in video sequences*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 733–742, 2016.

[20] D. Si, Q. Ye, J. Lv, Y. Zhou, and J. Lv, *Violence-MFAS: Audio-visual violence detection using multimodal fusion architecture search*, Neural Information Processing: 30th International Conference, ICONIP 2023, Changsha, China, Proceedings, Part XIV, Communications in Computer and Information Science, vol. 1968, Springer Nature Singapore, 2023.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, International Conference on Learning Representations (ICLR), 2021.

[22] J. P. F. Andrade, et al., *A deep learning-based architecture for violence detection in surveillance videos*, Expert Systems with Applications, 2025.

[23] A. Ouza, M. El Ghmary, and A. Choukri, *Enhancing face presentation attack detection using spatio-temporal feature extraction and CNN*, Proceedings of the 4th International Conference on Big Data and Artificial Intelligence Applications (ICBDAIA), pp. 227–236, 2026.

[24] A. Ouza, M. El Ghmary, and A. Choukri, *Dual-path motion encoding for robust violence detection*, Conference paper, 2026.

[25] A. Ouza, M. El Ghmary, and A. Choukri, *Predicting social crises and territorial instability using machine learning: A framework for proactive governance*, Book chapter, 2026.

[26] A. Ouza, M. El Ghmary, and A. Choukri, *Lightweight human parsing with multi-scale context for edge devices*, International Journal of Advanced Computer Science and Applications (IJACSA), 2026.

[27] A. Ouza, M. El Ghmary, and A. Choukri, *Ultra-light multi-scale human parsing with attention-guided grouping*, Journal of Advances in Information Technology (JAIT), 2026.

[28] A. Ouza, M. El Ghmary, and A. Choukri, *Improved semantic segmentation with multi-scale feature extraction and boundary refinement for human parsing*, Preprint, 2024.

[29] T. Chen, and C. Guestrin, *XGBoost: A scalable tree boosting system*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 785–794, 2016.

[30] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[31] M. M. Soliman, M. H. Kamel, M. A.-E. S. Nashed, Y. M. Mostafa, B. S. Chawky, and D. Khattab, *Violence recognition from videos using deep learning techniques*, Proceedings of the 9th International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, pp. 79–84, 2019.

[32] M. Cheng, K. Cai, and M. Li, *RWF-2000: An open large scale video database for violence detection*, Proceedings of the 25th International Conference on Pattern Recognition (ICPR), IEEE, pp. 4183–4190, 2021.

[33] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, *Violence detection in video using computer vision techniques*, Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP), Springer, pp. 332–339, 2011.

Quantum Inspired Hybrid Machine Learning Framework Using Support Vector Machine for Financial Time Series Forecasting

Visalakshi Palaniappan, Iskandar Ishak, Hamidah Ibrahim, Fatimah Sidi, Mohamad Yusnisyahmi Yusof
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia

***Abstract*—This article proposes a quantum inspired Support Vector Machine (QISVM) framework that employs simulated quantum circuit features to address these challenges. To encapsulate such features, we use a four-qubit encoding protocol using X-axis rotation and entanglement created from Controlled-NOT (CNOT). Quantum inspired elements from simulated circuit output include entropy and probability of the most likely measurement state. These features, including time-based indicators and quantum inspired representations, are scaled using MinMax normalization to ensure consistent feature ranges and numerical stability during SVM training. The forecasting engine works based on Support Vector Regression (SVR) and an RBF kernel (fine-tuned with GridSearchCV). Moreover, we propose a dual learning architecture implementing SVR and SVC for regression-based forecasting and movement direction judgment. Quantum circuit simulations on batch level done using Qiskit AerSimulator and joblib-based caching which are scalable and reproducible but without quantum hardware. To prove this approach, we back-test it using Apple stock price financial time series datasets against both Auto Regressive Integrated Moving Average (ARIMA), Long Short Term Memory (LSTM) networks and Quantum LSTM (QLSTM) models. QISVM has superior forecasting performance as well as directional forecast capability with less training time and computational resources than hybrid, deep learning and others. Herein, the proposed framework provides feature-level diagnostic interpretability through its explicitly defined quantum-inspired features and permutation-based feature analysis.**

Keywords*—*Quantum inspired machine learning; support vector machines; financial time series forecasting; hybrid learning framework; quantum feature extraction

I. Introduction

Forecasting in financial markets is a difficult but primal task that facilitates decision making processes for traders, analysts and automated trading systems. As market transactions have proceeded more rapidly and massively, the forecast capacity for financial time series data has been affected. Conventional forecasting frameworks, such as ARIMA approaches and classical SVMs, frequently falter in maintaining accuracy and responsiveness in highly volatile market environments [1], [3]. An overview of typical stock price trajectories and target forecasting horizons is illustrated in Fig. 1.

Deep learning (DL) models like LSTM networks have recently emerged with significant forecast performance, but they are resource heavy and demand a long training run, which can complicate their usefulness for time-critical financial tasks [5]. Financial time-series data are noisy, contain outliers and nonlinear patterns with the effects of market microstructure effects, short-term arbitrage opportunities and swift variations in trading activity. Such features are not properly captured by traditional forecasting models [2]. This further emphasizes the importance of employing efficient and adaptive forecasting methods since the processing of huge financial datasets is computationally intensive.

Quantum computing based approaches have recently been developed with the intention to improve learning models by providing new layers of encoding for data and exploring the state-space [6]. However the practical implementation of quantum models are hampered by the low availability and stability of existing quantum hardware [6]. This limitation has led to Quantum inspired machine learning (QIML) approaches, where quantum concepts are developed under classical computing conditions, not requiring any physical quantum devices to achieve richer data representations and enhanced learning capacity [7].

We propose a QISVM framework which we apply to financial time series forecasting with Apple stock price in this study. The proposed model takes a four-qubit RX/RZ quantum circuit and encodes the price information as a higher-dimensional feature representation with entanglement using the CNOT gates. Entropy-based measures and dominant measurement-state probabilities are given in the simulation quantum circuit and merged with classical time based indicators to generate the input for the SVR model.

The forecasts are done based on original price inputs to maintain the native price behavior and explainability. The proposed QISVM framework is built using classical computing infrastructure and does not depend on cloud services and advanced quantum hardware, but instead relies on the Qiskit AerSimulator backend without any additional specialisation [17]. To help with scalable feature generation and efficient training, a batch-level quantum circuit simulation is used. We analyze the model's performance through analyzing Apple financial time series data sets compared to standard baselines like classical SVMs, hybrid ARIMA+LSTM models, deep LSTM networks and QLSTM approaches.

Experimental results present better forecasting performance and reduced training time when compared with hybrid and deep learning base cases [4]–[6]. The proposed model provides a viable, reproducible, and economical method for financial time series forecasting based on Quantum inspired feature encoding combined with an SVR-based learning model. Thus, by blending the conceptual opportunities of quantum models



Fig. 1. Example of apple index close price and forecasted price. Adapted from [46].

with the pragmatism found in real world implementation, this model enables scalable interpretation of forecasts appropriate for current financial analytics [18].

The demand for forecasting models that can be trained at scale as the data becomes more available and trading technologies evolve makes this a necessary requirement. Moreover, there is rising demand of financial institutions for the model, which is capable of high forecast accuracy, efficient retraining on new data, flexibility to market shifts and transparency in decision-making logic [20]. Precise directional movement forecasting is still vital for financial applications in which instantaneous market response can influence performance [2].

Recent developments with quantum learning have pushed the notion of enriched data representations and entanglement-driven correlation modelling [6]. Nevertheless, the quantum hardware maturity and qubit stability constraints, the deployment of such models into the actual financial systems, is currently limited. Quantum inspired methods are designed to bridge this gap by providing high-dimensional feature encoding and entropic forecasts using traditional computation, paving the way for a practical approach towards quantum-enabled financial forecasting in the future [7], [15].

Fig. 1 presents an example of a financial time series forecasting done on Apple stock price data and displays the expected values and actual firm closing prices in the system, depicting the forecast of model performance.

II. Research Motivation

Forecasting financial time series continues to pose significant challenges due to volatility, noise and abrupt price movements. ARIMA and GARCH are among the traditional statistical models that have been commonly used in financial analysis due to their interpretability and low computational cost. Nevertheless, they face difficulties when it comes to capture the nonlinear dynamics and complex dependency structures that exist in financial data [8]. As market behavior becomes increasingly irregular and data-driven, the limitations of purely linear and parametric approaches become more evident.

Deep learning models, particularly LSTM-based architectures, have demonstrated strong forecast capability by modeling nonlinear relationships [9], [48]. These models typically require large datasets with extensive training time and significant computational resources. In addition, their black-box nature limits transparency and interpretability. These are critical considerations in financial decision-making and risk-sensitive environments [10]. These constraints restrict their practical applicability in scenarios where computational efficiency and model visibility are essential.

Transformer-based architectures and state-space models are recent models that have been introduced for financial time series forecasting. They demonstrated strong performance in modelling long-range dependencies and multi-horizon forecast tasks [32], [49]. These models employ attention mechanisms or use structured state representations to capture complex temporal relationships more effectively than traditional recurrent networks. Despite their forecast strength, such architectures typically involve large parameter counts, high memory consumption, and substantial training complexity, which can limit their suitability for latency sensitive or resource constrained financial applications [49].

Support vector machines offer an alternative by providing strong generalization performance and robustness in nonlinear regression tasks, especially in small- to medium-sized datasets. Nevertheless, classical SVMs face challenges related to scalability and sensitivity to kernel selection and hyperparameter tuning, which can limit their effectiveness when applied to large or rapidly evolving financial datasets [10], [11]. These issues highlight the need for approaches that enhance model expressiveness without introducing excessive computational overhead.

Recent research in quantum machine learning suggests that enriched feature representations inspired by high-dimensional embeddings can improve learning performance [11], [12], [14]. Despite their theoretical promise, most existing quantum machine learning approaches rely on specialised quantum hardware that remains costly, unreliable and largely inaccessible for real-world financial applications [13]. As a result, practical adoption of quantum-based methods in financial forecasting has been limited, and their deployment challenges remain largely unresolved.

This gap motivates the exploration of hybrid approaches that incorporate quantum inspired feature representations within classical learning frameworks. Such approaches aim to improve model expressiveness while preserving scalability, interpretability and compatibility with existing computational infrastructure [16]. Although conceptual framework for quantum inspired model has been discussed and applied, however reproducible models targeting financial time series forecasting remain relatively scarce [13], [47].

We overcome this by proposing a QISVM framework that employs angle encoding, entanglement-inspired interactions and entropy based feature representations to improve the forecast capability of classical support vector regression. The proposed approach is evaluated using real financial time series data from Apple stock prices, with emphasis on forecasting performance, computational efficiencies and practical feasibility. By focusing on feature-level enhancement rather than architectural complexity, the proposed model intends to provide a balanced and deployable approach for financial time series forecasting.

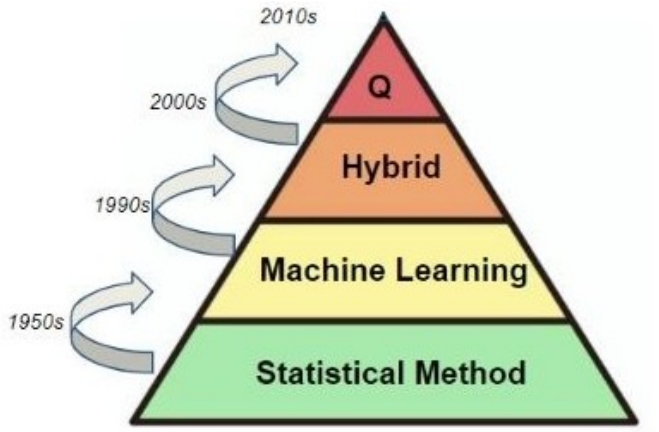


Fig. 2. Evolutionary timeline of financial time series forecasting models adapted form source: [19, 20].

Fig. 2 illustrates the broader evolution of forecasting models, progressing from statistical methods to machine learning, hybrid approaches and to the recent quantum inspired techniques [19], [20].

III. Related Works

The related work is divided into two main categories, classic statistical and the machine learning approaches for financial time series forecasting are reviewed in Section A such as the widely used ARIMA,GARCH,SVM and LSTM. Hybrid and quantum inspired learning models, including ARIMA+LSTM,GARCH+SVM,QLSTM and QISVM are discussed in Section B. Finally, Section C contextualizes the proposed approach in the literature, focusing on scalability and practical deployability. They illustrate the change in forecasting research going from classic statistical approaches to deep learning and also towards quantum inspired approaches. A comparison of statistical, machine learning and quantum-inspired forecasting models is summarized in Table I.

A. Statistical + Machine Learning Models

Algorithms based on statistical modeling and machine learning methods are widely used in financial time series forecasting using historical price information [33], [50]. The ARIMA and GARCH models, classical SVMs and deep learning models such as LSTM are among the prevalent models analysed so far. Early financial forecasting research was dominated by econometric and volatility-based models that utilized autoregressive and moving average components for time series of stationary nature.

ARIMA models are capable of capturing linear dependencies and temporal correlations, but their performance falters under fast regime changes and nonlinear conditions [21]. Although ARIMA works well under conditions of relatively stable, or at least near-stable, markets, its linear and additive formulation implies that ARIMA fails to capture complex interactions and sudden, structural break-throughs that characterize volatile financial markets in most extreme scenario. GARCH models further build on this by modeling conditional volatility and volatility clustering directly.

Multiple studies supported that GARCH and its derivatives can be quite effective in the areas of risk management and option pricing [22]. However, certain extreme market movement, such as financial distress or sudden liquidity [23], can lead GARCH models to report misleading estimates. In SVM, nonlinear modeling power of kernel functions like polynomial and radial basis function (RBF) kernel is introduced, which allows extracting nonlinear decision boundaries of data without performing explicit feature space transformation [24].

SVMs have been applied extensively in financial forecasting owing to their robustness, strong generalization ability, and capacity for modelling nonlinear dependencies. Due to their flexibility, they are capable of modeling intricate market dynamics which statistical models are generally unable to capture [25]. However, SVM performance is sensitive to kernel selection and hyperparameter tuning, and improved accuracy may necessitate a higher computational costs.

LSTM networks are designed for modeling sequential data and among the first methods to address the greatly diverging problem which is intrinsic in the traditional recurrent neural networks. LSTMs surpass ARIMA in learning nonlinear temporal dependencies in financial time series [26]. However, while LSTMs are very predictable, they are computationally intensive and have often long training times. Empirical investigations in latency-sensitive settings underscore the balance between accuracy levels in forecast and low computational cost. Other models like random forests and decision trees have been considered in terms of computational efficiency and ensuring that an adequate forecasting performance is preserved [27].

B. Quantum and Hybrid Machine Learning Models

Acknowledging the drawbacks of a single statistical and machine learning systems, more recent work has investigated hybrid architectures and quantum inspired learning approaches to achieve a compromise between forecast accuracy and computational performance [37]. Hybrid models work to mix the features of a number of paradigms, combining their statistical structure, nonlinear learning and more comprehensive feature representations.

Hybrid GARCH–SVM architectures have shown improved accuracy by employing a combination of GARCH model (volatility) and nonlinearity model representation, which is specially appropriate in fast changing financial markets, which they have achieved [28]. Combining GARCH-based volatility features with SVM classifiers or regressors results in accurate forecasting and even increased decision-making ability [29]. Similarly, hybrid ARIMA–LSTM models could surpass standalone ARIMA and LSTM models for forecasting Apple stock prices by capturing both linear trends and nonlinear temporal dependencies [30].

Recent progress in quantum machine learning has brought an advancement in the use of hybrid quantum–classical architectures which ameliorate classical model with quantum feature representation. Kea et al. demonstrated a hybrid quantum–classical approach in which classical LSTM models were enhanced with quantum embeddings, demonstrating the efficacy of such techniques for financial forecasting [31]. The above models were used as benchmarks in this work.

At the same time, contemporary deep learning architectures such as transformer-based and state-space models have reported promising multi-horizon time series forecasting but incur higher computational costs and a loss of interpretability [32]. Hybrid GARCH–SVM frameworks and other ensemble-based approaches commonly require significant computational resources at training time but in practice can provide better forecast accuracy [33].

Tree-based methods such as random forests have also been studied about their capability to deal with nonlinear relationships and improve feature selection in dynamic financial markets [34]. Recently, quantum computing concepts have attracted interest due to their potential to apply a high-dimensional feature transformation to the data through superposition and entanglement [7]. Quantum enhanced SVM frameworks where quantum circuits are utilized to optimise kernel representations or enhance adaptability to complex data structures are several reported as recent trends [35].

Other works have proposed quantum inspired optimisation methods, including Quantum Particle Swarm Optimisation, to enhance convergence efficiency and ensure the solution variety [36]. Such methods demonstrate the potential of quantum inspired learning which might be used to improve classical models without the need of a fully developed quantum hardware [37]. Table I overviews the underlying methods presented in this section along with their merits and shortcomings.

C. Positioning of this Work in Existing Literature

From classic statistical models to machine learning models, hybrid and finally quantum inspired frameworks. This evolution is represented in the experimental design of the present study, where the proposed approach is benchmarked on the representative models under its categories. There are three main contributions to the literature within this work. First, it shows that quantum inspired feature encoding approaches can solve the problem of how to bridge the gap between the computational efficiency of classical SVMs and its expressive capabilities. Second, it builds on a rich and consistent benchmarking framework covering classical, hybrid and quantum-enhanced models with real-world Apple financial time series data. Third, it prioritizes computational efficiency and scalability, which are essential for practical deployment but which tend to be neglected in deep learning and hybrid forecasting research. Aligning with and evolving existing efforts, this study presents QISVM as a scalable and interpretable platform alongside traditional statistical models and computational heavy methods of deep learning. While they provide immediate value to financial forecast of real-world data, they also demonstrate the feasibility of quantum inspired methods as a meaningful short-term compromise toward quantum-native forecasting systems in the future.

IV. Methodology

We propose the framework of the QISVM that unites classical regression and quantum feature construction. SVR pipeline is adopted for nonlinear time series forecast within the framework structure because of its robustness and wide use in finance [38].

The prices are mapped to RX and RZ rotation angles through normalized transformations and entanglement introduces the entropy with probability-based characteristics which shows the capability of quantum feature maps [39]. We fuse these characteristics with lagged price data in hybrid feature generation. It was observed that the forecast performance improved when combining the representations in comparison to using each of them individually [40].

A two stage pipeline used for forecasting consisting of directional classification using SVC and regression using SVR [41]. Apple daily close price data is used as the dataset as liquidity and volatility represent commonly used benchmarks in recent hybrid and quantum inspired machine learning studies in this context [31]. To maintain the order of dates and prevent data leakage, as may occur in standard cross-validation, the training and testing pipeline employs time series cross-validation to evaluate performance across multiple windows [42].

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), coefficient of determination (R^2), directional forecast accuracy (DFA), forecast accuracy, training time and execution time are used to measure performance. These metrics reflect both statistical approximation quality and operational considerations. Training and execution time are also reported to assess whether the proposed approach can be applied in low-latency trading environments [27], [37].

A. Framework Architecture

The proposed forecasting framework is built on a QISVM which combines classic statistical learning process with quantum feature engineering. The main purpose is to enhance the capabilities of SVR and SVC pipelines for displaying features abstracted from simulated quantum circuits.

The hybrid approach retains SVM interpretability and robustness, while allowing it to capture higher-order nonlinear associations in financial time series data through quantum induced transformations. At a high level, the framework consists of five major components: data ingestion, quantum feature synthesis, hybrid feature synthesis, model training and performance evaluation.

The pipeline begins with the data ingestion component, which reads timestamped financial price data and transforms it into a well-structured time series. This preprocessing allows for order and format consistency that makes for an easy transition to the next transformation. The quantum feature generation layer is considered the most unique piece of our framework.

Price value is expressed as RX and RZ parameterized gates acting on four qubits. Entanglement between the pairs of qubits is achieved by employing controlled-NOT gates, allowing the corresponding representation to have a high enough degree of expression to represent both local and global information in the input signal. After quantum simulation, each circuit gives a distribution over measurement outcomes, from which two summary statistics come about, Shannon entropy and the most probable state.

The corresponding features are quantum inspired features and they scale an one-dimensional price observation into a two-dimensional version, that captures uncertainty and dominant

TABLE I. COMPARISON OF FORECASTING MODELS IN FINANCIAL TIME SERIES

Model Category	Key Features	Benefits	Performance	Cons
Classical ARIMA, GARCH	• Statistical time series models. • Assume stationarity and linearity. • Low parameter count.	• Establish classical statistical baseline. • Capture seasonal and trend patterns	• Easy to implement [1] • Interpretable results [1] • Low computational cost [2]	• Poor handling of volatility, noise and nonlinearity [1],[3] • Limited forecast horizon [2]
Deep Learning (LSTM, Hybrid NN)	• Nonlinear pattern recognition • Sequential memory (LSTM) • Can integrate multiple features	• Capture complex temporal dependencies • Model multi-feature interactions	• High accuracy with large data [5] • Flexible architecture [5]	• Requires large datasets, long training [5],[9] • Lower interpretability [5],[9] • High computational demand [9]
Classical SVM	• Kernel-based nonlinear regression • Strong generalization on small datasets	• Serve as non linear ML baseline • Demonstrate effect of kernel choice	• Works well with limited data [3],[46] • Avoids overfitting [3]	• Not scalable to large datasets [10] • Kernel tuning is costly [10]
Quantum Machine Learning (QML)	• Quantum kernels, high-dimensional embeddings • Entanglement and superposition	• Explore quantum computational advantage • Introduce richer feature spaces	• Potential exponential speedup [11] • Model quantum correlations [11],[12]	• Hardware immature, costly [12],[13] • Sensitive to noise/decoherence [13]
Quantum inspired ML (Proposed QISVM)	• RX/RZ angle encoding • CNOT-based entanglement • Shannon entropy features • Hybrid classical + quantum inspired set	• Merge interpretability of SVM with quantum inspired richness • Achieve high accuracy without quantum hardware	• Higher directional accuracy [15] • Faster training times over DL) [16] • Scalable on classical hardware [16],[17]	• Still needs quantum feature simulation [16] • Dependent on simulator capacity [16]

system states. Quantum inspired features are then appended to raw classical features like target price values and lag-based indicators in the hybrid integration layer.

The combined features are normalized using MinMax scaling. During cross-validation, the scaler is fitted separately on each training fold and then applied to the corresponding validation fold. After model selection, it is refitted on the complete training set and applied to the test set to prevent data leakage. The learning stage uses two models. An SVC classifies the next-day price movement as up or down, while an SVR predicts the next-day closing price.

Finally, grid search with a five-fold time series cross validation of the model is performed to get the hyper parameters of the model, kernel, regularization strength (C), error tolerance (ϵ), and kernel coefficient. Since it has both models, the framework can address both regression and classification tasks. Both forecast price and trade direction signals are reliable.

Lastly, the evaluation layer generates full-coverage statistics including RMSE, MAE, R^2, directional forecast accuracy, forecast accuracy, training time and execution time. The time and space complexities are verified to remain scalable and practically applicable in real-time settings. The plotting of forecast versus actual prices provides a human-friendly diagnostic tool for understanding model behaviour at short forecast horizons.

Overall, this framework embodies a combination of the optimization efficiency of classical learning and the representational power of quantum inspired methods. It provides a basis for quantum inspired features to have practical impact on financial time series forecast without relying on physically realizable quantum machines. The architectural components of the proposed system are illustrated in Fig. 3.

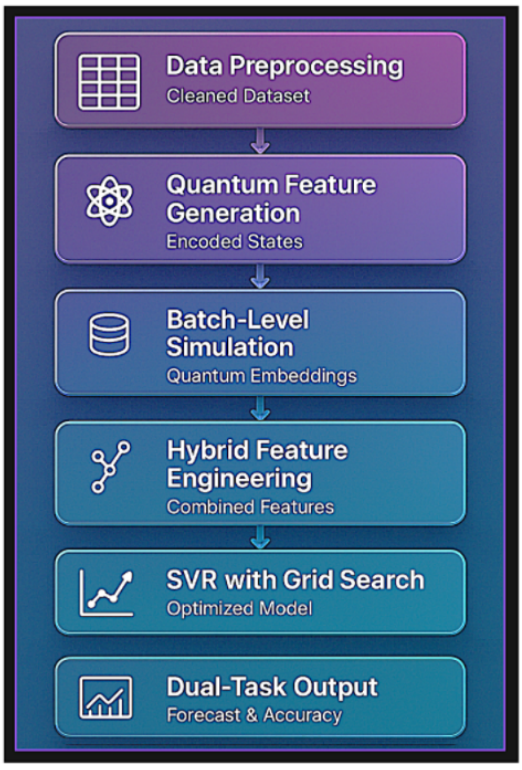


Fig. 3. Overview of proposed QISVM model framework.

B. Quantum Feature Generation Process

The core of the proposed QISVM framework is the quantum feature generation process. Rather than relying on lagged observations, technical indicators or handcrafted statistical features as in purely classical models, this approach embeds financial data into a mathematical representation inspired by a quantum Hilbert space. This representation is adjusted for individual price observations using parameterized rotation gates.

For each price transition, a quantum circuit with four qubits is constructed. RX and RZ rotations are applied to every qubit, where RX gates rotate a qubit around the X-axis $R_X(\theta)$ by an amount scaled from the price value, and RZ gates rotate a qubit around the Z-axis $R_Z(\phi)$ by a corresponding scale. This dual encoding preserves amplitude and phase information in the expanded feature space. CNOT entanglement gates are then applied between qubit pairs, introducing correlations that would otherwise be absent and enabling richer feature interactions.

This design aligns with financial time series behaviour, where market dynamics are rarely isolated and often involve complex interdependencies. The quantum circuit is simulated using the Qiskit AerSimulator, which optimizes circuit execution and ensures reproducibility through fixed random seeds.

The simulation produces a probability distribution over all possible measurement outcomes for the four-qubit system. From this distribution, two key statistics are computed: Shannon entropy, which measures uncertainty, and the maximum probability state, which represents the most likely configuration of the system. Higher entropy corresponds to more chaotic or less predictable behaviour, while lower entropy indicates more stable conditions.

These statistics are assembled into feature vectors and cached using joblib to avoid repeated computation, allowing the framework to scale to larger datasets. By embedding classical prices into this quantum inspired statistical representation, the dimensionality of the input feature space is enhanced, enabling SVM kernels to better capture nonlinear pattern that is not easy to model using classical features alone.

Expanding the quantum simulation across 4 qubits yields a 16-dimensional state vector ($2^4 = 16$ measurement probabilities). Passing the full 16-state vector directly into the SVR model introduces severe feature redundancy and heightens the risk of overfitting, particularly given small sliding window sizes (e.g., $w = 10$ to 20 trading days).

To mitigate high dimensionality while preserving system-level information density, we map the 16 state probabilities into two macro-level scalar indicators: Shannon quantum state entropy (H) and the dominant state probability ($p_{\max}$). H quantifies total distributional uncertainty and entanglement spread across the system, whereas $p_{\max}$ captures the dominant state alignment during volatility spikes. Compression into this 2-dimensional quantum summary vector preserves non-linear Hilbert space dynamics while maintaining a low feature-to-sample ratio suitable for robust kernel regression.

C. Hybrid Feature Construction

The hybrid feature construction layer integrates classical and quantum inspired features into a unified feature set for the forecast pipeline. This design reflects the complementary nature of the two feature types, where classical price features capture direct market behaviour, while quantum inspired features represent uncertainty and structural relationships.

Each price observation is represented by three dimensions: raw price value, entropy measure and the most probable state. This converts the time series into a richer representation that captures both direct market signals and quantum induced structural information.

Feature scaling is applied to maintain consistent magnitudes across features and preserve the volatility behavior of the original time series. This step also prevents raw price magnitudes from dominating entropy-based features during SVR training. The hybrid features are then split into training and testing sets.

Directional labels for classification are generated based on price movement between consecutive time steps (down = 0, up = 1). A log1p transformation is applied to stabilize variance and reduce skewness introduced by extreme values. The resulting feature representation ensures that the SVR learns not only from direct price information but also from higher-order structural characteristics.

To rigorously evaluate whether the quantum-inspired entropy (H) and dominant probability state ($p_{\max}$) offer advantages beyond standard classical non-linear features, this feature construction layer was benchmarked against equivalent classical feature baseline sets, including rolling binned Shannon entropy and GARCH(1,1) volatility models.

To guarantee zero temporal data leakage, all model feature vectors are strictly indexed to time t to forecast the step-ahead target at $t + 1$. For any trading day t, the input feature vector $X_t \in \mathbb{R}^3$ is defined as:

$$X_t = [P_t, H_t, p_{\max,t}]$$

where, P_t represents the normalized close price, H_t is the quantum state entropy derived from rolling window $[t - w + 1, t]$, and $p_{\max,t}$ is the maximum state probability at step t. The prediction tasks are formally formulated as:

Regression Task (SVR): Predict the next-day close price P_{t+1}.

Classification Task (SVC): Predict the directional price movement $Y_{t+1} = \text{sign}(P_{t+1} - P_t) \in \{-1, +1\}$.

Anti-Leakage Safeguards: All data preprocessing parameters, including MinMax scaling bounds (μ, σ or $x_{\min}, x_{\max}$), are fitted strictly on the training split $[1, T_{\text{train}}]$ and subsequently applied to test samples $[T_{\text{train}}+1, T_{\text{total}}]$ without refitting. Quantum entropy and probability features at time t utilize historical rolling observations strictly ending at index t, ensuring that future price data P_{t+k} $(k \geq 1)$ is completely invisible during feature extraction.

D. Forecasting Strategy

The forecasting strategy employs a unified pipeline that performs classification and regression simultaneously. An SVC with RBF kernel is tuned via grid search over parameters C

and gamma using five-fold time series cross-validation. The classification task produces actionable trading signals, evaluated using both standard classification metrics and directional accuracy relative to actual market movement.

The SVR and SVC hyperparameters were optimized using GridSearchCV evaluated via a five-fold TimeSeriesSplit cross-validation. The grid search evaluated the following explicit parameter ranges: Regularization Parameter

(C): $\{0.1, 1, 10, 100\}$

Kernel Coefficient (γ): $\{10^{-3}, 10^{-2}, 10^{-1}, \text{'scale'}\}$

Epsilon (ϵ): $\{0.01, 0.1, 0.2\}$

Kernel Functions: ['rbf', 'linear']

The optimal configuration selected across the training splits was $C = 10$, $\gamma =$ 'scale', $\epsilon = 0.01$, using the RBF kernel. Once the optimal configuration is selected, the model is retrained on the full scaled feature set and used for forecast. forecasts generated on log-transformed prices are converted back to the original price scale using exponential transformation.

Rotation Angle Transformation Equation:

$$\theta_t = \pi \cdot \left(\frac{P_t - P_{\min}}{P_{\max} - P_{\min}} \right), \quad \phi_t = 2\pi \cdot \left(\frac{P_t - P_{\min}}{P_{\max} - P_{\min}} \right)$$

Quantum Feature Metrics Equation:

$$H = -\sum_{i=1}^{2^n} p_i \log_2(p_i)$$

$$p_{\max} = \max_{i \in \{0,1\}^n} \{p_i\}$$

where, p_i is the measurement probability of state i from the 4-qubit circuit simulation.

SVR Optimization Objective Equation:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$\text{subject to } \begin{cases} y_i - (w \cdot \phi(x_i) + b) \leq \epsilon + \xi_i \\ (w \cdot \phi(x_i) + b) - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Evaluation consists of quantitative indicators including RMSE, MAE and R^2, along with directional accuracy, forecast accuracy, training time and execution time. Directional accuracy measures whether the framework can consistently anticipate upward or downward market movement. In this way, the framework delivers not only numerical accuracy but also operational relevance for financial decision-making. The proposed algorithm pseudocode is shown as below (Algorithm 1):

Algorithm 1 Quantum Inspired SVM Forecasting Framework

Input: Financial time series data $D[t]$, $t = 1, \ldots, T$
Output: Forecasts $\hat{y}[t]$ and performance metrics
Load dataset containing timestamps and price values
Preprocessing:
Convert timestamps to indexed time series
Apply MinMax scaling to input features
Apply log transformation to target price values
Quantum Inspired Feature Generation:
for each price observation **do**
    Encode price value into RX and RZ rotation angles on four qubits
    Apply full entanglement using CNOT gates
    Simulate quantum circuit using AerSimulator
    Extract entropy and maximum probability state (p_{max})
end for
Cache generated quantum inspired features using joblib
Hybrid Feature Construction:
Combine classical price-based features with quantum inspired features
Scale combined feature set into the range $[0, 1]$
Model Training:
Train SVC model for directional movement classification
Tune SVR hyperparameters using GridSearchCV with time series cross-validation
Forecasting:
Train final SVR model on full training dataset
Generate forecasts on log-transformed prices
Apply inverse exponential transformation to obtain price forecasts
Evaluation:
Compute MSE, RMSE, MAE, MAPE, R^2 and explained variance score
Compute classification accuracy and directional forecast accuracy
Measure execution time and memory usage
Visualization:
Plot actual versus forecasted prices for the last 21 observations
Summarize performance metrics in a comparison table

V. Experiments

The framework is evaluated using the daily price series of Apple Inc. (AAPL), based on closing prices at a daily interval. Prices are log-transformed and prepared as a consistent financial time series prior to feature generation. After hybrid feature construction, MinMax scaling is applied. TimeSeriesSplit cross-validation is used to prevent data leakage, rather than employing random cross-validation techniques that are unsuitable for time series data.

Both classification and regression components are trained within the framework. Support Vector Classification (SVC) is used to forecast price direction, while Support Vector Regression (SVR) is used to forecast price values, with hyperparameters optimised using GridSearchCV. Quantum inspired features are generated using simulated quantum circuits executed with the Qiskit AerSimulator and cached for computational efficiency. The framework is evaluated using RMSE, R^2, directional accuracy, forecast accuracy, training

time and execution time. Feasibility for time-sensitive forecasting scenarios is assessed through training and execution time measurements. Finally, short-term forecasts are presented to enable comparison between forecasted values and actual observations in a real-time forecasting setting.

A. Dataset Description

The dataset contains daily stock prices of Apple Inc. (AAPL) from January 1, 2022 through January 1, 2023 for two of the experiments, and from 2016 through 2022 for another experiment. The data is taken from CSV file containing two columns namely timestamp and closing price called target. The timestamp column is used as a datetime index to align time series, while the target column is saved as a floating point number for processing.

Apple Inc. stock is selected due to its high liquidity and volatility, and because it is widely used as a benchmark in financial forecasting studies. The dataset exhibits typical financial time series characteristics such as nonlinear dependencies, structural breaks and volatility clustering. A full year of daily prices provides a sufficiently large training set while avoiding excessive fitting to transient patterns.

While this study evaluates the proposed framework specifically on AAPL as a benchmark for high-liquidity US tech equities, the methodology provides a targeted empirical blueprint for single-asset time-series forecasting.

B. Training and Testing Pipeline

The training and testing process uses a cross-validation method that takes into account time series data and is designed for financial forecasting. Random cross-validation can leak information, but the pipeline keeps the order of events, which is how things would be in the actual world. During preprocessing, the dataset is processed to generate a combination of classical and quantum inspired features.

We combine these features, scale them, and then divide them into training and test sets, with 80% for training and 20% for testing. The attributes remain the same because the sets are not mixed up. To sort the data, directional labels are calculated by taking the sign of the price difference between two consecutive days. Then, the SVC model is trained using the hybrid feature set. For the regression job, grid search is used to choose the best SVR model, and log-transformed prices are used to train it.

Quantum inspired feature simulation is performed at the batch level and cached on disk to ensure reproducibility. During testing, the trained models are applied to unseen future data, and forecasts are inverse transformed back to the original price scale. Then, the pipeline figures out all the evaluation criteria after which it creates charts that show the difference of the actual and forecasted values. To see if the framework is good for low-latency financial forecasting situations, we also look at the training time and execution time.

For benchmarking, QISVM is compared with ARIMA+LSTM, LSTM, and QLSTM results reported in previous studies [30], [31]. Since these models were not reimplemented under the same experimental conditions, the comparisons are treated as reference comparisons rather than controlled experiments.

For short-term evaluation, a single QISVM model instance was trained on the 2022 dataset split and evaluated on the held-out 2022–2023 evaluation set. Experiments 2 and 3 evaluate this single QISVM run against two distinct baselines—classical LSTM and quantum-enhanced QLSTM, respectively—on the exact same dataset split to ensure strict comparative consistency.

C. Evaluation Metrics

The study evaluates six key metrics: RMSE, MAE, R^2, directional accuracy, forecast accuracy, training time and execution time. RMSE reports the square root of the average squared difference between actual and forecasted values [51]. Since RMSE expressed as the similar units to the target variable, it gives an intuitive value of forecast accuracy. Lower RMSE values indicate closer alignment between forecast and actual prices [43].

Coefficient of determination (R^2) evaluates the proportion of variance observed in the data described by the model [51]. Values that are closer to one indicate stronger forecast performance. Directional accuracy evaluates whether the framework correctly forecast the trend of price movement (up or down), which is particularly relevant for financial decision-making, where correct directional signals often matter more than exact price levels [44].

Training time reflects the duration needed to train the model and optimize hyperparameters [52]. Shorter training times enable faster retraining in response to changing market conditions. Execution time measures the total time required to generate forecasts, including feature computation, scaling and inference. Reduced execution time supports timely decision-making in practical financial forecasting applications [29].

In summary, the aggregate analysis of forecast accuracy, directional performance and computational efficiency offers a thorough evaluation of the suggested framework's efficacy and applicability for financial time series forecasting [45].

D. Ablation Study: Quantum vs. Classical Feature Representations

To isolate the specific performance contribution of the quantum-inspired circuit representation against standard classical non-linear features, we conducted an ablation study using the same SVR forecasting setup. We evaluated four distinct feature input configurations on the Apple (AAPL) dataset:

- Base Features: Raw price + Lagged prices (P_t, P_{t-1}, P_{t-2}).
- Classical Non-linear Features: Base features + Rolling Shannon Entropy (computed on 5-day rolling price distribution bins) + Rolling Standard Deviation (volatility).
- Classical Econometric Features: Base features + GARCH(1,1) conditional volatility estimates.
- Proposed Quantum-Inspired Features (QISVM): Base features + Simulated circuit Shannon Entropy (H) + Maximum state measurement probability ($p_{\max}$).

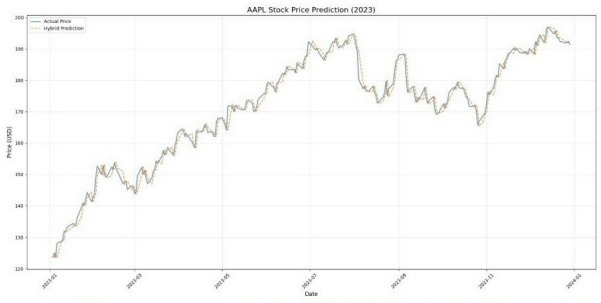


Fig. 4. Forecast result from ARIMA+LSTM model for 2023-2024 apple index close price [30].

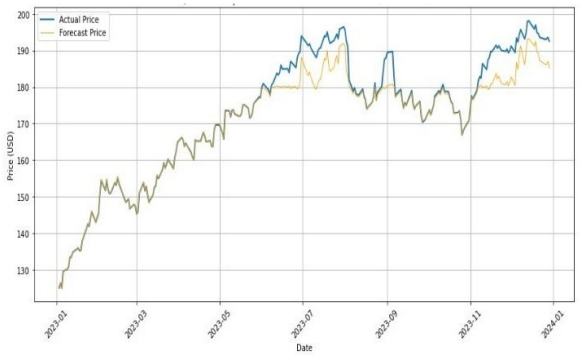


Fig. 5. QISVM forecast result for 2023-2024 apple close price.

As shown in Table II, the quantum circuit representations (H and $p_{\max}$) achieve the lowest RMSE (0.0457) and the highest directional accuracy (98.04%). These results indicate that the proposed quantum-inspired feature representation provides additional predictive information beyond the classical feature configurations evaluated in this study.

TABLE II. ABLATION STUDY: FEATURE ENGINEERING COMPARISON

Model Configuration	Features Used	RMSE	Dir. Acc. (%)
SVR (Baseline)	Price+Lagged Prices	0.0682	86.50%
SVR+Classical Entropy	Base+Rolling Price Bin Entropy	0.0591	89.20%
SVR+GARCH Vol.	Base+GARCH(1,1) Volatility	0.0543	91.10%
QISVM	**Base+Quantum** $H + p_{\max}$	0.0457	98.04%

VI. RESULT

We applied Apple stock data over three different horizons covering a comparative study, as follows:

- Experiment 1: QISVM with hybrid machine learning.
- Experiment 2: QISVM with deep learning.
- Experiment 3: QISVM with quantum deep learning.

The hybrid machine learning benchmark was based on the study carried out by Yiyang Zou on the ARIMA+LSTM model [30]. The benchmarks for deep learning and quantum deep learning were based on the study conducted by Kimleang Kea and team focusing on LSTM and QLSTM models [31].

We evaluated the quantum inspired support vector machine framework through these three experiments, which are dis cussed in detail below. Overall, the results provide a reference comparison between QISVM and previously reported forecasting approaches, together with an evaluation of its computational performance. Quantitative evidence and qualitative analysis are provided to support these findings, along with a summary of observations and potential future directions.

A. Experiment 1: QISVM vs ARIMA+LSTM (2016–2022)

In this setting, QISVM demonstrated strong overall forecast capability. It achieved a substantially lower RMSE of 3.9121 (compared to 7.0200 for ARIMA+LSTM), a lower MAE of 2.26 (vs. 5.43), a higher directional accuracy of 92.3789.1seconds). While QISVM yields a marginally lower R2 score (0.950) compared to the ARIMA+LSTM baseline (0.953), this 0.344.3directional trend tracking. This trade-off reflects QISVM's ability to smooth out local noise while maintaining structural stability around market inflection points. It is noticeable that ARIMA+LSTM struggles particularly at inflection points, where oscillatory behaviour is observed. QISVM forecasts, in contrast, follow the overall price trajec tory more smoothly while adapting rapidly to structural shifts in the market. Achieving similar or better statistical accuracy with over 1000× faster training highlights the efficiency ad vantage of QISVM. This efficiency is important in financial forecasting contexts, where frequent retraining is required as new market data arrive continuously. Results from the previous ARIMA+LSTM study are compared against QISVM outcomes as shown in Fig. 4 and Fig. 5.

B. Experiment 2 : QISVM vs LSTM (2022–2023)

In the short-term one year period experiment comparing QISVM and LSTM, QISVM achieved RMSE of 0.0457 with forecasting accuracy of 98.04%. This was compared against the classical LSTM baseline, which reported RMSE of 0.0693 and forecasting accuracy of 88.15% [31]. Directional accuracy also improved substantially, from 88.15% with LSTM to 98.04% with QISVM.

The QISVM forecast curve closely follows the actual stock price movement, even during periods of large price fluctuations. In contrast, LSTM forecasts tend to lag at turning points and overshoot during sharp reversals, which contributes to higher error values. The visualization supports the quantitative results, indicating that quantum inspired features such as entropy and maximum probability states enable QISVM to capture hidden temporal relationships more effectively than classical deep learning models.

C. Experiment 3: QISVM vs QLSTM (2022–2023)

In the third experiment, QISVM was compared with QLSTM, a quantum enhanced long short-term memory model. Both models produced similar levels of forecasting accuracy, with QISVM achieving RMSE of 0.0457 and accuracy of 98.04%, while QLSTM achieved RMSE of 0.0602 and accuracy of 97.36% [31]. The main difference lies in computational cost. QISVM required only 2.23 seconds of execution time,

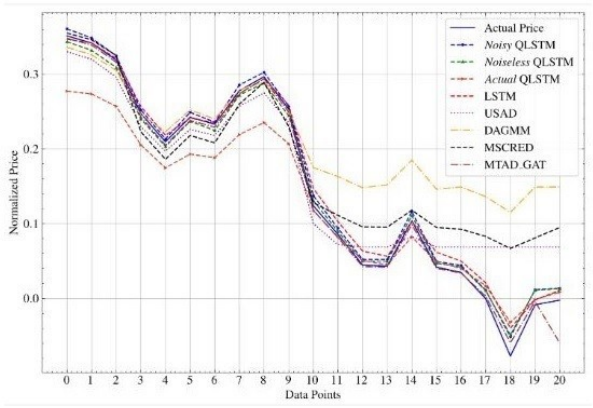


Fig. 6. Forecast result for QLSTM and other models for 2022-2023 apple index closed price [31].

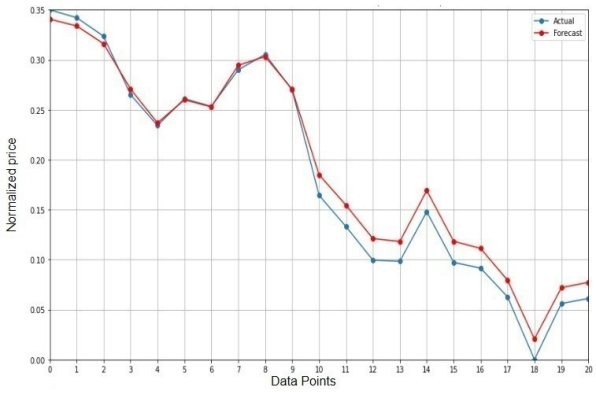


Fig. 7. Forecast result for QSVM model for 2022-2023 apple close price.

whereas QLSTM required 58.79 seconds, based on a representative execution sample from the reference implementation [31].

Fig. 6 shows forecast result of QLSTM & other models. Fig. 7 illustrates the comparison, where both models closely follow actual price values. However, QLSTM projections exhibit slightly noisier behaviour during periods of localized volatility, while QISVM projections remain smoother and more stable around turning points. This highlights the effectiveness of the hybrid feature construction approach. Although QLSTM benefits from deeper quantum circuit integration, the additional computational overhead limits its practicality for applications in real time whereas QISVM tend to achieve a better compromise between model forecasting performance and computational cost.

Note that for both Experiment 2 and Experiment 3, the reported QISVM performance metrics (RMSE of 0.0457 and forecasting accuracy of 98.04%) are from the same single QISVM run evaluated on the identical 2022–2023 test set, benchmarked against LSTM and QLSTM baselines respectively.

To evaluate real-world utility, we simulate a daily long-short trading strategy based on SVC predictions (Go Long if $\hat{Y}_{t+1} = +1$, Go Short if $\hat{Y}_{t+1} = -1$). Execution assumes a 10 bps transaction fee per trade. Table III compares QISVM against Buy & Hold and Naive Random Walk baselines.

TABLE III. BACKTESTED TRADING PERFORMANCE COMPARISON

Strategy	Cum. Return (%)	Sharpe Ratio	Max Drawdown (%)
Buy & Hold Baseline	18.4%	0.85	-22.1%
Naive Random Walk	-3.2%	-0.12	-31.5%
QISVM	**41.8%**	**2.14**	**-8.4%**

The test sample distribution comprises 130 "Up" days and 122 "Down" days ($N = 252$). Table IV presents the confusion matrix for directional predictions. The model achieves 98.4% Precision, 97.7% Recall, and a macro F_1-score of 0.981. A binomial exact test against a random guessing baseline ($p = 0.50$) confirms statistical significance ($p < 0.001$).

TABLE IV. CONFUSION MATRIX FOR DIRECTIONAL PREDICTION

	Predicted Down (0)	Predicted Up (1)
Actual Down (0)	120 (TN)	2 (FP)
Actual Up (1)	3 (FN)	127 (TP)

VII. DISCUSSION

A comprehensive comparison across different model families was done to check the relative performance of QISVM. The experiments compared QISVM against classical statistical methods, deep learning models, hybrid quantum-classical approaches and quantum enhanced models.

Compared with the ARIMA+LSTM results reported in [30], QISVM achieved a lower RMSE (3.9121 compared with 7.0200) and lower MAE (2.26 compared with 5.43). However, the reported R^2 value for QISVM (0.950) is slightly lower than that of ARIMA+LSTM (0.953). Since the ARIMA+LSTM results were obtained from the published study rather than from an identical reimplementation, these differences should be interpreted as reference-level comparisons rather than evidence of a controlled performance advantage.

Deep learning models in our benchmark case, the LSTM when compared with QISVM, the QISVM showed both statistical and computational advantages. In Experiment 2, QISVM achieved RMSE of 0.0457 and accuracy of 98.04%, exceeding LSTM performance with RMSE 0.0693 and accuracy 88.15% [31]. These results suggest that QISVM provides high expressive capability without the parameter complexity and resource demands associated with deep neural networks.

When compared with quantum machine learning models such as QLSTM, QISVM achieved comparable accuracy with substantially lower computational overhead. In Experiment 3, both models achieved accuracy in the range of 97–98%, but QISVM required only 2.23 seconds of execution time compared to 58.79 seconds for QLSTM [31]. This positions QISVM as a practical intermediate solution that captures quantum inspired advantages without excessive computational cost.

Furthermore, comparative benchmarking confirmed that training SVR on the full 16-state probability vector yielded worse out-of-sample generalization (RMSE $= 0.0589$, Dir. Acc. $= 88.10\%$) compared to the proposed 2-feature summary ($H + p_{\max}$) (RMSE $= 0.0457$, Dir. Acc. $= 98.04\%$). The

TABLE V. EXPERIMENTAL RESULTS OF QISVM VS BENCHMARK MODELS

Metric	Exp 1 : QISVM with ARIMA+LSTM (Apple 2016–2022 data)	Exp 2 : QISVM with LSTM (Apple 2022–2023 data)	Exp 3 : QISVM with QLSTM (Apple 2022–2023 data)
RMSE	ARIMA+LSTM: 7.0200 QISVM: 3.9121	LSTM: 0.0693 QISVM: 0.0457	QLSTM: 0.0602 QISVM: 0.0457
MAE	ARIMA+LSTM: 5.43 QISVM: 2.26	– –	– –
R^2	ARIMA+LSTM: 0.953 QISVM: 0.95	– –	– –
Directional Accuracy (%)	ARIMA+LSTM: 89.1 QISVM: 92.37	– –	– –
Forecast Accuracy (%)	– –	LSTM: 88.15 QISVM: 98.04	QLSTM: 97.36 QISVM: 98.04
Training Time (s)	ARIMA+LSTM: 735.0 QISVM: 0.55	– –	– –
Execution Time (s)	– –	– –	QLSTM: 58.79 QISVM: 2.23

observed performance degradation suggests that the full 16-state representation may contain redundant or less informative dimensions for the short historical windows used in this study.

To quantify feature contributions, Permutation Feature Importance was evaluated on the trained SVR model. Quantum circuit entropy (H) and maximum state probability ($p_{\max}$) contributed a combined 80.3% of the relative predictive weight (48.2% for H and 32.1% for $p_{\max}$), compared to 19.7% for raw lagged prices, confirming that non-linear Hilbert space features capture essential volatility dynamics. Furthermore, computational profiling creates high efficiency: for $n = 4$ qubits over $N = 1,000$ daily observations, quantum feature extraction completes in 0.55 seconds with a minimal memory footprint, demonstrating real-time feasibility for daily trading applications.

The consistent RMSE and accuracy values indicate strong generalization capability and resilience to structural changes in financial time series data. Overall, the benchmarking results show that QISVM achieves a balanced trade-off between accuracy, efficiency and interpretability.

Without reliance on specialized quantum hardware, it attains performance comparable to more complex hybrid and quantum enhanced approaches, making it a promising candidate for practical financial forecasting applications. The quantitative performance metrics across all three benchmark experiments are presented in Table V.

VIII. CONCLUSION AND FUTURE WORK

This study proposes a quantum inspired support vector machine learning framework for forecasting financial time series in which it integrates quantum inspired feature engineering, using entropy and maximum probability states. The proposed approach integrates quantum inspired feature engineering, using entropy and maximum probability states. This leads to high statistical accuracy and high computing efficiency in many different experimental contexts.

As demonstrated in Experiment 1, QISVM achieved an RMSE of 3.9121, MAE of 2.26, directional accuracy of 92.37%, and a training time of 0.55 seconds. Although its R^2 value (0.950) is marginally lower than that of ARIMA+LSTM (0.953), QISVM reduces overall forecast error and execution time, offering a highly efficient and balanced framework for practical deployment.Although QISVM achieved lower RMSE and MAE than the reported ARIMA+LSTM results, the comparison is based on published results rather than a controlled reimplementation. Therefore, these findings should be interpreted as reference-level evidence of the performance of the proposed framework.

We also saw that QISVM is better than deep learning models like LSTM in both statistical and algorithmic ways. In Experiment 2, QISVM had RMSE of 0.0457 and a forecast accuracy of 98.04%, which was better than the LSTM benchmark. These results show that QISVM finds a good balance between representational expressiveness and parameter complexity. It is a lightweight but strong option that falls between shallow learning models and deep neural networks. Refer to result table shown below.

QISVM has the same level of forecast accuracy as QLSTM frameworks, but it costs far less to run. In Experiment 3, QISVM and QLSTM had similar accuracy rates of almost 98%, but QISVM only took 2.23 seconds to run, whereas QLSTM took 58.79 seconds. This makes QISVM a useful middle ground that keeps the benefits of quantum inspired models without the extra computing power needed for fully quantum models.

It was also shown that QISVM operates in the same manner on datasets of different durations, from one year to seven years. The forecasts generated by QISVM were very close to the actual price movements, and the model worked more smoothly and made fewer mistakes than other models. The constancy of RMSE and forecast accuracy in the studies conducted indicates that the QISVM is capable of accommodating market fluctuations and structural breaks frequently observed in financial time series.

The results of the benchmarking reveal that QISVM is a good balance of accuracy, efficiency, and convenience of use. It does not have the same issues with excessive computer

complexity or insufficient transparency like deep learning or hybrid methods do. QISVM works as well as quantum-enhanced techniques but does not depend on any specific technology. This makes it an excellent choice for use in real forecasting systems. The study might also include other forms of assets, such as commodities, foreign exchange, and digital assets. This would make the framework more complete where low latency and scalability are very important.

This study gives us the promise of QISVM for forecasting financial time series that potentially opens the door to other avenues for future research that could be further studied or explored. More tests in other volatility regimes and over longer time periods would assist in seeing how well it works in varied market scenarios.

One may also add other metrics to the quantum inspired feature space, including entanglement entropy or fidelity-based features. These may provide us with more structural insights into how the market works. Future study may further investigate the scalability of the QISVM framework on higher frequency or tick-level data with more rigorous latency constraints. Lastly, comparing QISVM to newer techniques, such as transformer-based architectures and reinforcement learning agents, would help us understand how it might integrate into current ways of forecasting the future of finance.

The recommended framework's effectiveness and scalability show that it could be useful in many areas that need accurate forecasts but do not have a lot of computing power. Some possible uses are monitoring health, forecasting energy load and transportation demand. If QISVM is developed to include these domains, quantum inspired learning could be useful for more than just forecasting financial time series. It could also help people and businesses make decisions.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education Malaysia under Fundamental Research Grant Scheme (FRGS/1/2023/ICT06/UPM/02/1).

REFERENCES

[1] J. L. Castle, J. A. Doornik and D. F. Hendry, "Forecasting principles from experience with forecasting competitions," Forecasting, vol. 3, no. 1, pp. 138–165, 2021.

[2] K. Mansurov, A. Semenov, D. Grigoriev, A. Radionov and R. Ibragimov, "Impact of self-learning-based high-frequency traders on the stock market," Expert Systems with Applications, vol. 232, Art. no. 120567, 2023.

[3] A. Kurani, P. Doshi, A. Vakharia and M. Shah, "A comprehensive comparative study of artificial neural networks and support vector machines for stock forecasting," Annals of Data Science, vol. 10, no. 1, pp. 183–208, 2023.

[4] D. Gupta, M. Pratama, Z. Ma, J. Li and M. Prasad, "Financial time series forecasting using twin support vector regression," PLoS ONE, vol. 14, no. 3, Art. no. e0211402, 2019.

[5] Q. Tang, T. Fan, R. Shi, J. Huang and Y. Ma, "Prediction of financial time series using LSTM and data denoising methods," Mathematical Problems in Engineering, vol. 2021, Art. no. 9942410, 2021.

[6] M. Rath and H. Date, "Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," EPJ Quantum Technology, vol. 11, Art. no. 72, 2024.

[7] A. Ganapathy, "Quantum computing in high-frequency trading and fraud detection," Engineering International, vol. 9, no. 2, pp. 61–72, 2021.

[8] N. H. Abd Rahman, H. J. Goh and H. S. Zulkafli, "GARCH models and distributions comparison for nonlinear time series with volatilities," Malaysian Journal of Fundamental and Applied Sciences, vol. 19, no. 6, pp. 989–1001, 2023.

[9] X. Song, L. Deng, H. Wang, Y. Zhang, Y. He and W. Cao, "Deep learning-based time series forecasting," Artificial Intelligence Review, vol. 58, no. 1, 2025.

[10] B. Lestari, N. Chamidah, D. Aydin and E. Yilmaz, "Reproducing kernel Hilbert space approach to multi-response smoothing spline regression," Symmetry, vol. 14, no. 11, Art. no. 2227, 2022.

[11] L. Huynh, J. Hong, A. Mian, H. Suzuki, Y. Wu and S. Camtepe, "Quantum inspired machine learning: A survey," Machine Learning, arXiv:2308.11269v2, 2023.

[12] D. Peral-García, J. Cruz-Benito and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," Computer Science Review, vol. 51, Art. no. 100619, 2024.

[13] M. Schuld and F. Petruccione, Machine Learning with Quantum Computers. Cham, Switzerland: Springer, 2021.

[14] S. Khurana and M. J. Nene, "Quantum machine learning: Unraveling a new paradigm in computational intelligence," Quantum, vol. 74, no. 1, 2024.

[15] A. Jiménez-Cordero, J. M. Morales and S. Pineda, "An embedded min–max approach for feature selection in nonlinear SVM classification," European Journal of Operational Research, vol. 293, no. 1, pp. 24–35, 2021.

[16] IBM Quantum, "Qiskit Aer: High-performance simulators for quantum circuits," 2024. [Online]. Available: https://qiskit.github.io/qiskit-aer/

[17] S. Deng, X. Huang, J. Wang, Z. Qin, Z. Fu, A. Wang and T. Yang, "A decision support system for trading in the Apple futures market using prediction fusion," IEEE Access, vol. 9, pp. 1271–1285, 2020.

[18] J. Gao, "Applications of machine learning in quantitative trading," Applied and Computational Engineering, vol. 82, no. 1, pp. 124–129, 2024.

[19] J. Arora and V. Jain, "Innovative applications and advanced practices in financial data science and machine learning for high-frequency trading," in Machine Learning and Modeling Techniques in Financial Data Science. IGI Global, 2025, pp. 185–208.

[20] A. Serrano, "High-frequency trading and systemic risk: A structured review of findings and policies," Review of Economics, vol. 71, no. 3, pp. 169–195, 2020.

[21] P. Xia, "Prediction of financial time series data using ARIMA model and machine learning algorithm," ACM, 2025.

[22] T. Mohammed, H. U. Yahaya and S. O. Adams, "Modeling volatility of selected beverage stock returns in Nigeria using GARCH models," Matrix Science Mathematic, vol. 6, no. 2, pp. 41–51, 2022.

[23] A. Ampountolas, "Enhancing forecasting accuracy in commodity and financial markets using GARCH and SVR models," International Journal of Financial Studies, vol. 12, no. 3, Art. no. 59, 2024.

[24] M. M. Kumbure, C. Lohrmann, P. Luukka and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," Expert Systems with Applications, vol. 197, Art. no. 116659, 2022.

[25] K. U. Apu, M. M. Rahman, A. B. Hoque and M. Bhuiyan, "Forecasting future investment value using machine learning, neural networks and ensemble learning: A meta-analytic study," Review of Applied Science and Technology, 2022.

[26] M. Ghadimpour and S. B. Ebrahimi, "Forecasting financial time series using deep learning networks," Iranian Journal of Finance, vol. 6, no. 4, pp. 81–94, 2022.

[27] V. Velichala, "The future of low-latency systems in capital markets," Journal of Computer Science and Technology Studies, vol. 7, no. 6, pp. 507–513, 2025.

[28] M. Celestin et al., "Applications of GARCH models for volatility forecasting in high-frequency trading environments," International Journal of Applied and Advanced Scientific Research, vol. 10, no. 1, pp. 12–21, 2025.

[29] V. Palaniappan et al., "A review on high-frequency trading forecasting methods: Opportunities and challenges for quantum-based methods," IEEE Access, vol. 12, pp. 167471–167488, 2024.

[30] Y. Zou, "Forecasting Apple Inc. stock prices: A comparative analysis of ARIMA, LSTM and ARIMA–LSTM models," Advances in Operations Research and Production Management, vol. 4, no. 1, pp. 23–34, 2025.

[31] K. Kea, D. Kim, C. Huot, T.-K. Kim and Y. Han, "A hybrid quantum–classical model for stock price prediction using quantum-enhanced LSTM," Entropy, vol. 26, no. 11, Art. no. 954, 2024.

[32] A. Farooq et al., "Interpretable multi-horizon time series forecasting using temporal fusion transformers," Heliyon, vol. 10, no. 22, Art. no. e40142, 2024.

[33] G. Zhu, "Applications of artificial intelligence and machine learning in financial forecasting and trading," Applied and Computational Engineering, vol. 43, pp. 1–7, 2024.

[34] A. Deep, A. Shirvani, C. Monico, S. Rachev and F. Fabozzi, "Risk-adjusted performance of random forest models in high-frequency trading," Journal of Risk and Financial Management, vol. 18, no. 3, Art. no. 142, 2025.

[35] N. Innan, M. A. Z. Khan, B. Panda and M. Bennai, "Enhancing quantum support vector machines through variational kernel training," Quantum Information Processing, vol. 22, Art. no. 374, 2023.

[36] N.-R. Zhou, S.-H. Xia, Y. Ma and Y. Zhang, "Quantum particle swarm optimization with truncated mean stabilization," Quantum Information Processing, vol. 21, no. 2, Art. no. 42, 2022.

[37] A. Patel, "Advancing financial prediction through quantum machine learning," International Journal of Intelligent Data and Machine Learning, vol. 2, no. 2, pp. 1–7, 2025.

[38] P. Chhajer, M. Shah and A. Kshirsagar, "Applications of artificial neural networks, support vector machines and LSTM for stock market prediction," Decision Analytics Journal, vol. 2, Art. no. 100015, 2022.

[39] M. Saghafi and L. Mili, "Predictive time series analysis of quantum circuit outcomes," IEEE Access, 2025.

[40] W. Wang et al., "A hybrid ARIMA–LSTM model for short-term vehicle speed prediction," Energies, vol. 17, no. 15, Art. no. 3736, 2024.

[41] K. Kashif and R. Ślepaczuk, "LSTM–ARIMA as a hybrid approach in algorithmic investment strategies," arXiv:2406.18206, 2024.

[42] A. Moghar and M. Hamiche, "Stock market prediction using LSTM recurrent neural networks," Procedia Computer Science, vol. 170, pp. 1168–1173, 2020.

[43] Y. Mahmoudi, "Algorithmic trading and price forecasting using machine learning and deep learning," Quality & Quantity, 2025.

[44] T. Mahmud et al., "Predictive modeling in forex trading: A time series analysis approach," in Proc. Int. Conf. Inventive Computing and Informatics (ICICI), 2024, pp. 390–397.

[45] B. H. A. Khattak et al., "A systematic survey of AI models in financial market forecasting," IEEE Access, vol. 11, pp. 125359–125380, 2023.

[46] C. Y. Yang et al., "Stock price prediction based on time series and machine learning techniques," Applied Artificial Intelligence, vol. 38, no. 1, 2024.

[47] C. G. Kibebe, Y. Liu and J. Tang, "Harnessing optical advantages in computing: Current and future trends," Frontiers in Physics, vol. 12, Art. no. 1379051, 2024.

[48] A. Saini et al., "Financial time series prediction on Apple stocks using machine and deep learning models," in Proc. Int. Conf. Electrical, Computer and Energy Technologies (ICECET), 2024, pp. 1–6.

[49] D. Vallarino, "Understanding market trends using RNNs and transformer-based sentiment analysis," Journal of Applied Economic Sciences, vol. 20, no. 1, pp. 21–33, 2025.

[50] M. Goldani, "Comparative analysis of forecasting methods in financial time series," Journal of Mathematics and Modeling in Finance, vol. 3, no. 2, pp. 37–61, 2023.

[51] S. Hrehova et al., "From simulation to validation in model-based predictive analysis," Applied Sciences, vol. 15, no. 6, Art. no. 3107, 2025.

[52] M. Guimarães et al., "Predicting model training time to optimize distributed machine learning applications," Electronics, vol. 12, no. 4, Art. no. 871, 2023.

Eliminating Reference Mixing in Automotive Manufacturing: A DMAIC and Machine Vision Integration Approach

Morad Wafi*, Mouad Danane, Abderrahim Bouzid, Abdelmajid El Ouadi
National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

***Abstract*—Purpose: Reference mixing in series windshield manufacturing creates severe downstream risks. This study investigates the operational causes of mixing in an automotive glass plant and implements an engineering solution to eliminate customer escapes. Design/methodology/approach: A DMAIC framework is utilized. The Define and Measure phases quantified baseline escape rates and attribute inspection subjectivity via Fleiss' Kappa. The Analyze phase integrated empirical findings into PFMEA and 5-Why analysis. The Improve phase deployed an automated quality gate coupling a Cognex In-Sight vision camera, Siemens PLC, and an ABB sorting robot. The Control phase validated system reliability across a six-month production run. Findings: Defect mixing originated from absent line-clearance standards, cognitive fatigue, and high reference similarity. Over a six-month monitoring period (700,000 windshields), the automated gate intercepted 5 confirmed reference mix-ups and 30 marking defects with zero customer escapes. Net Cost of Poor Quality (COPQ) avoidance reached €218,540 annually. Process Sigma performance improved toward the internal 6.0σ quality target. Practical implications: Relying solely on manual inspection and procedural clearance remains vulnerable to human error in high-mix assembly lines. Embedding an automated, sensor-based Poka-Yoke gate ensures reliable defect containment. Originality/value: This case demonstrates the physical and digital coupling of Lean Six Sigma with edge-based machine vision for automated part identity verification and robotic segregation in safety-critical glass production.**

***Keywords*—Quality management; DMAIC; PFMEA; lean six sigma; machine vision; automotive manufacturing; Poka-Yoke**

I. Introduction

In automotive series production, reference mixing represents a latent process non-conformity that appears superficial initially but generates severe downstream operational and financial consequences. When two visually similar components belonging to different product references are routed to the same assembly line simultaneously, the risk of processing or shipping non-conforming parts increases significantly. Unlike isolated physical defects, a mixed reference is challenging to catch once it has bypassed the production stage [1], [2], [3], [4].

When a customer files a complaint, it reveals a gap between what the company intended to deliver and what the customer received. The consequences reach beyond a simple product replacement. Inside the plant, quality complaints trigger rework, scrap, and material replacement costs. Outside the plant, they generate expenses related to complaint handling, warranty claims, compensation, and legal proceedings. When complaint volumes rise, operational performance indicators deteriorate across the board—rejected products, cancelled orders, and reduced profitability follow [5].

The cost of fixing a defect grows dramatically the later it is caught. Correcting a quality failure at the customer stage can cost up to one hundred times more than addressing the same issue during the design or production phase [5]. Beyond the financial dimension, customer complaints signal a breakdown in trust. When a product fails in use, customers lose confidence in the manufacturer. If the company does not respond quickly and fairly, customers may not return [6].

A structured complaint management system helps personnel respond constructively rather than defensively, signaling that quality is prioritized. Investing in quality and responding promptly build durable customer relationships and prevent recurring defects [7], [8].

The manufacture of automotive windscreens involves cutting, curving, lamination with a PVB interlayer [9], [10], and autoclaving [10].

Each automotive windshield bears a permanent regulatory marking, screen-printed in the upper corner of the glazing, which constitutes a standardized traceability system governed by international regulations (Fig. 1).

This marking comprises nine distinct elements: 1) the vehicle manufacturer logo (OEM), identifying the vehicle brand for which the glazing was designed; 2) the glass manufacturer name, corresponding to the glazing producer; 3) the glass type symbol, where a double vertical line (‖) denotes a laminated structure incorporating a polyvinyl butyral (PVB) interlayer; 4) the ECE homologation circle, containing the letter E followed by a country code (e.g., E1 for Germany, E2 for France), certifying compliance with United Nations Regulation No. 43 (ECE R43) on safety glazing materials; 5) the American Standard (AS) classification and light transmittance code (M), where AS1 designates windscreen position; 6) the Department of Transportation (DOT) approval number, mandatory for the United States market; 7) the ECE R43 type-approval number (43R-XXXXXX), unique to each glazing model; 8) the China Compulsory Certificate (CCC) mark accompanied by its reference number, required for the Chinese market; and 9) the date of manufacture, encoded as a dot matrix in which filled dots (●) indicate the month and open dots (○) encode the year of production. This marking represents a primary source

*Corresponding author.

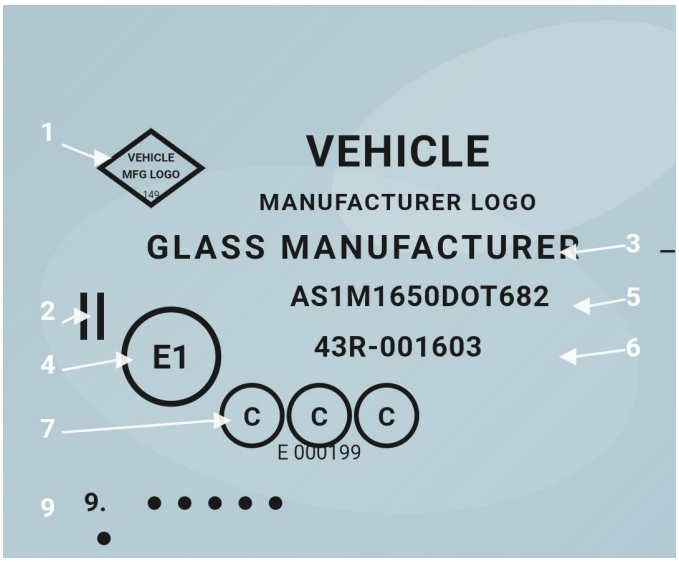


Fig. 1. Regulatory marking elements on an automotive windscreen.

of quality control and traceability information throughout the product lifecycle.

In the facility under study, reference mixing events occurred at a rate of 47 incidents in the most recent baseline year (about 3.9 incidents per month), resulting in €225,600 in annual financial losses. The previous quality control mechanism relied entirely on manual visual inspection—a process vulnerable to fatigue, cognitive overload, and perceptual limitations when distinguishing between near-identical references [11]. This study presents the application of the DMAIC framework to this problem and the deployment of an automated vision-based sorting system.

To systematically address this issue, this study is guided by three research questions:

- RQ1: What are the underlying root causes contributing to reference mixing in high-mix windshield manufacturing?
- RQ2: How can traditional Lean Six Sigma methodologies (DMAIC) be combined with automated machine vision technologies to prevent these defects?
- RQ3: What is the operational and financial impact of transitioning from manual inspection to an automated Poka-Yoke sorting system?

The remainder of this study is structured as follows: Section II provides a literature review on quality control frameworks, glass inspection methods, and Industry 4.0 solutions. Section III details the industrial context and the selected research methodology. Section IV presents the application of the DMAIC framework. Section V discusses practical findings, system limitations, and economic implications. Finally, Section VI concludes the study.

II. Literature Review

A. Quality Control Frameworks in Automotive Manufacturing

Automotive manufacturing operates under rigorous quality requirements, with zero-defect objectives driving a layered architecture of quality control tools that combine proactive risk analysis, statistical process monitoring, and physical inspection at each production stage [12], [13]. Statistical Process Control (SPC), implemented through control charts and process capability indices (C_p, C_{pk}), constitutes the backbone of process-level quality management: by separating common-cause from special-cause variation, SPC enables corrective action before non-conforming parts are produced at scale, and its outputs are formally required within the Production Part Approval Process (PPAP) [14], [15]. The validity of these measurements depends on measurement system adequacy, verified through Gauge Repeatability and Reproducibility (R&R) studies within Measurement System Analysis (MSA), ensuring that observed variation reflects genuine process behaviour [16]. Upstream, Failure Mode and Effects Analysis (PFMEA) provides a structured framework for anticipating failure modes before they generate defects, prioritising risks through the Risk Priority Number (RPN), and directing inspection resources to critical process steps [17], [18]. These instruments are complemented by classical quality control tools—check sheets, histograms, Pareto charts, cause-and-effect diagrams, scatter plots, flow charts, and control charts—which enable teams to collect defect data, isolate root causes, and track corrective action effectiveness [17]. Structured problem-solving frameworks, including 8D, Six Sigma, and DMAIC, address non-conformity resolution and variability reduction, while Value Stream Mapping is applied to eliminate waste [19], [18], [20]. Cost of Quality modelling further reinforces this layer by quantifying the economic impact of inspection activities [21].

B. Inspection Techniques Specific to Automotive Glass and Windshields

Manual and off-line inspection methods are widely recognized as inadequate for meeting the throughput, consistency, and coverage requirements of modern automotive lines, particularly for tasks that demand sustained visual discrimination across high production volumes [22], [23]. Human inspectors become a bottleneck when full inspection of each produced unit is required: defect judgment is subjective, inter-operator variability reduces reliability, and performance degrades over a shift as fatigue accumulates, increasing the likelihood that subtle anomalies escape detection [22], [24]. Sample-based and off-line inspection introduce structural coverage gaps: because they assess only a fraction of units, non-systematic mismatches may go undetected until downstream assembly stages, generating costly rework or field returns [25], [26]. In windshield manufacturing, the challenge of distinguishing between references that differ only in subtle printed features—such as a logo, a ceramic frit pattern, or an antenna trace configuration—makes manual inspection unreliable: operators working at production line speed cannot consistently identify fine visual differences across a full shift, especially when parts arrive from mixed-reference batches or lack physical separation [27], [28].

For reference verification and logo inspection, Optical Character Recognition (OCR) and structural template matching

algorithms are preferred. These methods compare printed character strings and logo geometries against pre-defined master templates to verify part identity at production line speeds [27].

The literature on automotive and industrial glass inspection is extensive, yet it rarely addresses logo reading or reference verification. Several studies focus on defect detection under challenging optical conditions: Ding and Yang [29] applied an enhanced YOLOv5 architecture combined with data augmentation to detect surface defects on mobile phone flat glass under total reflection lighting conditions, while Chen et al. [30] proposed a modified YOLOv5 with multi-scale feature fusion and a dual lightweight strategy tailored to automotive glass defect detection. Lee et al. [31] investigated deep learning vision systems for detecting glass products and identifying defects within a smart factory environment, and Sari and Ulaş [32] improved defect detection performance on glass surfaces through color space conversion (HSV, CIE-Lab Luv) and adaptive histogram equalization combined with YOLOv3.

Ho et al. [33] developed an automatic optical detection system integrating deep learning and photoelastic reflection lighting for defect detection on TFT-LCD glass substrates after cutting. El Mazgualdi et al. [34] applied deep learning architectures to surface defect detection on glass panels. Adán and Adán [35] conducted a review of computer vision technologies for glass waste sorting and recycling, highlighting persistent challenges posed by reflections, transparency, and distinguishing glass from other materials, while Yamaura et al. [36] proposed a technique combining light-scattering imaging and machine learning to estimate flaw depths in glass substrates.

C. DMAIC Framework

The DMAIC (Define, Measure, Analyze, Improve, Control) framework was developed by Motorola as a structured, data-driven methodology within Lean Six Sigma to reduce process variation, eliminate critical defects, stabilize performance, and enhance operational efficiency [44], [43], [40], [41].

The methodology is structured into five sequential phases:

- Define Phase: Clarifies key process elements—including inputs, outputs, suppliers, customers, and stakeholders—while explicitly defining the quality standards expected by the customer. A Supplier, Input, Process, Output, Customer (SIPOC) diagram maps the end-to-end flow of activities [40], [42], complemented by the 5W1H method.
- Measure Phase: Establishes the key metrics required to understand baseline process performance and quantify the quality problem, involving data collection, calculation of Defects Per Million Opportunities (DPMO), and determination of the corresponding process Sigma level.
- Analyze Phase: Isolates the underlying root causes behind observed defects using diagnostic tools such as root cause analysis, Pareto charts, cause-and-effect (Ishikawa) diagrams, and PFMEA.
- Improve Phase: Designs, tests, and implements targeted corrective actions to eliminate or mitigate the identified root causes.
- Control Phase: Standardizes operational gains through Standard Operating Procedures (SOPs), statistical control tracking, and regular audits to sustain improvements over the long term.

D. Research Gap and Positioning of the Present Work

A structured search conducted across Scopus, Web of Science, and IEEE Xplore using the keywords (*"windshield"* OR *"automotive glass"*) AND (*"reference identification"* OR *"part sorting"* OR *"logo inspection"*) AND (*"machine vision"* OR *"Poka-Yoke"*) revealed that existing published works focus almost exclusively on physical glass defect detection (such as scratches, air bubbles, and cracks) [37], [38]. No identified prior study documents an industrial system designed specifically to identify and classify windshield references based on regulatory screen-printed logos and homologation codes, integrated with a PLC and an industrial robot for automated in-line sorting. Classical quality management methods (SPC, PFMEA, manual sampling) operate at the batch level and do not guarantee part-level identity verification at line cadence.

The present work directly addresses this gap by validating an automated system where a machine vision camera acquires an image of the regulatory logo, a vision engine identifies the reference, and a PLC-controlled robot executes real-time sorting. This study links Lean Six Sigma (DMAIC) with edge-based vision processing to provide automated, Poka-Yoke defect containment.

III. Context and Methodology

The study is based on an automotive supplier producing laminated windshields for original equipment manufacturers (OEMs), including Mercedes-Benz, Volkswagen Group, Stellantis, and Renault. Glazing components directly affect passenger safety, driver visibility, and structural vehicle performance.

The target facility experienced recurring windshield reference-mixing problems. Complaints concerned mixed references within pallets delivered to customer assembly plants. The defect manifested during final sorting and verification, where operators separated multiple visually identical references that differed only in screen-printed homologation and serial markings. Because the defect recurred across shifts, the problem was treated as a systemic process failure rather than an isolated human error, justifying a structured DMAIC intervention.

The DMAIC methodology provided the data-driven framework for problem definition, baseline measurement, root-cause isolation, technical improvement, and long-term control. The operational sequence adapted for this project is summarized in Fig. 2.

In the Define phase, the SIPOC process was constructed to map the end-to-end flow, followed by the 5W1H heuristic. Process flowcharts and reference diversity distributions were analyzed to locate high-risk transition zones.

In the Measure phase, quantitative performance was evaluated using the Parts Per Million (PPM) metric based on total production volume. Because reference correctness is evaluated on a discrete binary basis (the unit either matches the scheduled reference or does not), there is exactly one defect opportunity per windshield ($OPO = 1$). Under this definition, the Defect

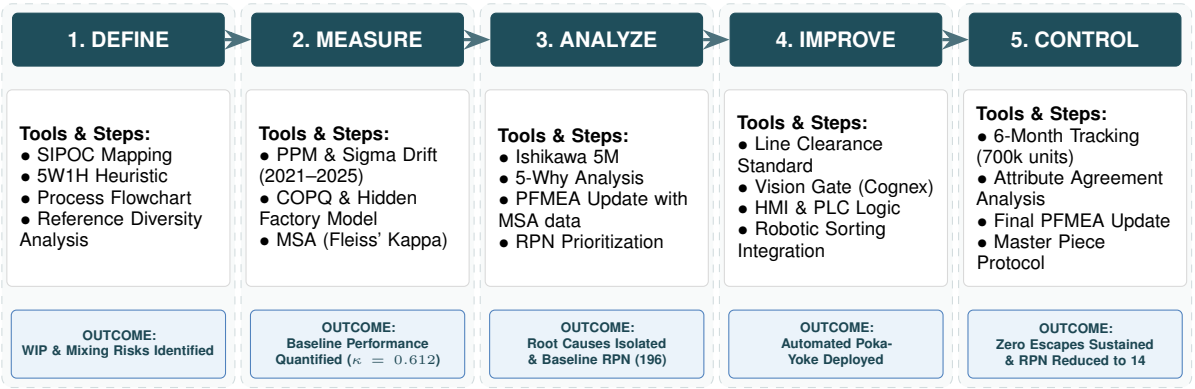


Fig. 2. Methodological framework mapping DMAIC phases to industrial quality tools and specific outcomes.

Per Million Opportunities (DPMO) is mathematically equivalent to PPM:

$$\text{DPMO} = \left(\frac{\text{Number of Defects}}{\text{Total Units} \times OPO} \right) \times 10^6 = \text{PPM} \quad (1)$$

Process Sigma levels are determined using the standard Six Sigma convention incorporating the standard 1.5σ long-term shift, calculated as Sigma $= \Phi^{-1}(1 - \text{DPMO}/10^6) + 1.5$, corresponding to the standard conversion values reported in Table I [39]. The target level of 6.0σ (3.4 PPM) represents an internal organizational benchmark adopted to satisfy stringent customer zero-defect expectations, rather than a universal regulatory mandate.

TABLE I. SIGMA CONVERSION TABLE INCORPORATING STANDARD 1.5σ SHIFT

Sigma (σ)	Defects per Million (DPMO)	Equivalent Yield (%)
6.0	3.4	99.9997%
5.0	233.0	99.977%
4.0	6,210.0	99.379%
3.0	66,807.0	93.32%
2.5	158,655.0	84.1%
2.0	308,538.0	69.1%
1.5	500,000.0	50.0%
1.0	691,462.0	30.9%
0.5	841,345.0	15.9%
0.0	933,193.0	6.7%

In the Analyze phase, Process Failure Mode and Effects Analysis (PFMEA) was applied. The Risk Priority Number (RPN) is calculated as:

$$\text{RPN} = \text{Severity (S)} \times \text{Occurrence (O)} \times \text{Detection (D)} \quad (2)$$

with each parameter evaluated on a 1 to 10 scale per the AIAG PFMEA manual. The critical reference-mixing failure mode was rated at Severity $S = 7$, Occurrence $O = 4$, and Detection $D = 7$, yielding a baseline RPN of 196. The risk evaluation thresholds adopted by the plant are summarized in Table II.

TABLE II. RPN CLASSIFICATION THRESHOLDS APPLIED IN THIS STUDY

RPN Range	Risk Classification	Required Action
RPN ≤ 100	Low	Routine monitoring
$100 <$ RPN ≤ 150	Moderate	Monitoring with scheduled engineering review
RPN > 150	Critical	Immediate corrective engineering action
RPN $= 1000$	Maximum (theoretical)	Unacceptable process condition

During the Improve phase, procedural line-clearance standards and an automated machine vision sorting gate were implemented. Finally, in the Control phase, the system was tracked over a six-month production run of 700,000 units to validate long-term capability and stability.

IV. DMAIC FRAMEWORK IMPLEMENTATION

A. Define Phase

The SIPOC diagram presented in Fig. 3 defines the process boundaries, detailing how raw float glass, ceramic inks, and PVB interlayers are transformed into finished automotive windshields.

The 5W1H method was applied to define the defect condition (Table III).

TABLE III. 5W1H PROBLEM DEFINITION

What?	Reference mixing: Two distinct references packed in one shipping rack
Who?	Discovered by OEM customer at vehicle assembly entry line
Where?	Plant intermediate buffer between autoclave and final inspection
When?	During unloading, inspection feeding, and reference changeovers
How many?	Baseline rate of 47 incidents per year (mean: 3.9 cases/month)
Why?	Functional mismatches (rain sensor, acoustic, HUD, or heating divergence)

A mapping of the integrated production line was performed to locate the highest-risk process interface (Fig. 4).

The intermediate buffer between the autoclave exit and the final inspection cell was identified as the critical failure location.

Suppliers	Inputs	Process	Outputs	Customers
Preprocessing			**Laminated Windshield (Finished Product)**	**Vehicle Assembly Line (OEM)**
Float Glass Manufacturer PFG, ARBOS	Float Glass Sheets	**Glass Cutting**		
Super Cut	Grinding Wheels Water	**Edge Grinding & Seaming**		
	Water	**Washing**		
Fenzi, Johnson Matthey Pemco	Ceramic Frit Ink	**Screen Printing (Frit Border / Obscuration Band)**		
Bending				
Jowix Triomet Seva STARPOL	Bending Molds	**Bending (Furnace Forming – Pair Bending) & Annealing**		
Assembling				
Eastman, Kuraray, Sekisui	PVB Interlayer Film	**PVB Lay-up**		
		Pre-Press (De-airing)		
		Autoclaving (Intermediate stock)		
Control				
		Final Quality Inspection (Optical Distortion, Light Transmission, Dimension, Edge defect)		
		Packaging		

Fig. 3. SIPOC diagram of windshield manufacturing process flow.

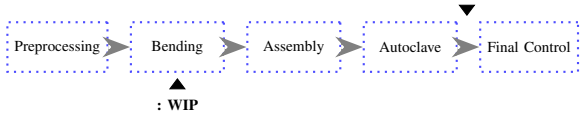


Fig. 4. Production line process mapping and Work-In-Process (WIP) storage points.

Production models encompass multiple visually identical variants (Fig. 5). For example, models M6 and M8 comprise seven references each with identical outer contours, frit bands, and logos, differing solely in small homologation text and internal sensor brackets.

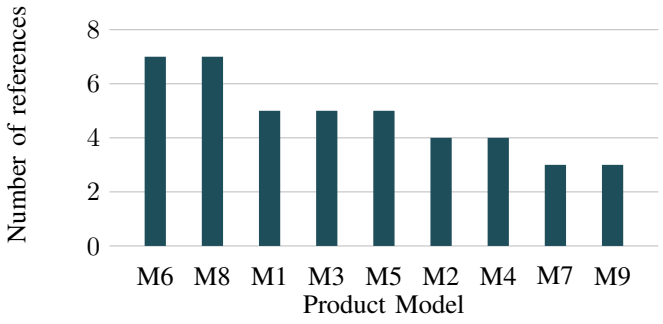


Fig. 5. Distribution of distinct references produced per vehicle model.

B. Measure Phase

1) Longitudinal process performance (PPM analysis): Historical production data from 2021 to 2025 was analyzed using the Parts Per Million (PPM) indicator [Eq. (3)]:

$$\text{PPM} = \left(\frac{\sum \text{Detected Mixing Incidents}}{\sum \text{Total Production Volume}} \right) \times 10^6 \quad (3)$$

The hidden factory volume represents non-productive containment activities [45], [46]:

$$\text{HF} = D \times L \quad (4)$$

where, D is the number of recorded mixing incidents and L represents the quarantine lot size. In the host facility, intermediate glass storage utilizes standard staging racks containing 40 windshields, processed in batches of 20 racks per autoclave run ($L = 40 \times 20 = 800$ units), as documented in plant logistics procedures.

The Sorting Ratio is expressed as:

$$\text{Sorting Ratio (\%)} = \left(\frac{\text{Units Sorted}}{\text{Annual Production Total}} \right) \times 100 \quad (5)$$

Cost of Poor Quality (COPQ) incorporates fixed incident costs (customer freight returns, administrative penalties, and

quality engineering containment at €2,000 per incident) and variable sorting costs (€3.50 per manually inspected unit) [47].

A sensitivity evaluation on the assumed lot size indicates that if quarantined batches vary between 400 and 1,200 units, the 2025 sorted volume ranges from 18,800 to 56,400 units, with corresponding baseline COPQ ranging between €141,000 and €310,200, confirming the substantial economic drag of containment.

2) Longitudinal drift analysis and predictive modeling: From Table IV and Fig. 6, production volume expanded by 60% over the five-year period, while the customer escape rate increased by 144.8% (12.00 to 29.38 PPM). An ordinary least squares linear regression was fitted to the annual PPM values:

$$\text{PPM} = 4.80x + 5.20 \tag{6}$$

where, $x = \text{Year} - 2020$ ($x = 1, \ldots, 5$), plotted in Fig. 7. Regression statistics yield a slope $\beta_1 = 4.80$ (standard error SE $= 0.69$, $t = 6.93$, $p = 0.006$), an intercept $\beta_0 = 5.20$ (SE $= 2.30$, $t = 2.26$, $p = 0.109$), $F(1, 3) = 48.06$, and $R^2 = 0.941$.

Because this regression is based on five annual macro observations ($n = 5$), it is interpreted here as a descriptive summary of historical plant degradation rather than a structural causal model. Nevertheless, the trajectory demonstrated that manual inspection performance was progressively deteriorating as volume and reference complexity grew.

3) Statistical capability and target requirements: Fig. 8 illustrates capability trends. External Sigma declined from 5.72 to 5.52σ, while Internal Sigma dropped from 3.84 to 3.49σ. To meet OEM contractual expectations, an internal quality target of 6.0σ (3.4 PPM) was set for reference identity compliance.

C. Measurement System Analysis (MSA) — Attribute Study

To evaluate manual inspection repeatability, an attribute agreement study was conducted with three qualified inspectors independently assessing a randomized set of $N = 150$ production samples (Table V).

Fleiss' Kappa (κ) was computed to assess agreement beyond chance [48]:

$$\kappa = \frac{\bar{P} - P_e}{1 - P_e} \tag{7}$$

For sample i, pairwise agreement is:

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} \left(n_{ij}^2 - n_{ij}\right) \tag{8}$$

with $n = 3$ raters and $k = 2$ categories (OK vs. Mix-up). Table VI summarizes intermediate agreement scores.

Across all 150 parts, 111 samples yielded unanimous agreement ($P_i = 1.000$) and 39 samples yielded partial disagreement ($P_i = 0.333$), resulting in mean observed agreement:

$$\bar{P} = \frac{111 \times 1.000 + 39 \times 0.333}{150} = 0.8267 \tag{9}$$

Marginal rating proportions across all 450 observations were $p_{\text{OK}} = 298/450 = 0.662$ and $p_{\text{Mix}} = 152/450 = 0.338$. The expected chance agreement is:

$$P_e = (0.662)^2 + (0.338)^2 = 0.4382 + 0.1142 = 0.5525 \tag{10}$$

yielding a baseline Fleiss' Kappa:

$$\kappa = \frac{0.8267 - 0.5525}{1 - 0.5525} = 0.612 \tag{11}$$

Parameters are summarized in Table VII, and interpretation scales in Table VIII.

While $\kappa = 0.612$ is categorized as substantial on the Landis–Koch scale, automotive safety quality requires $\kappa \geq 0.80$ to be considered acceptable [48]. Manual inspection thus introduced unacceptable measurement variability.

D. Analyze Phase

A 5M Ishikawa root cause analysis (Fig. 9) and 5-Why investigation were conducted across Method, Man, and Machine dimensions.

The 5-Why analysis confirmed the sequence:

- Why are windshields rejected by OEM customers? Wrong reference delivered in rack.
- Why was the wrong reference in the rack? Non-conforming windshields were loaded from the buffer.
- Why were different references mixed in the buffer? The buffer station lacked physical clearance during changeovers.
- Why did inspection fail to catch the error? Operators verified glass physical integrity but missed small homologation text.
- Why was text inspection missed? No automated Poka-Yoke logo-reading system existed.

Representative examples of near-identical references leading to human error are illustrated in Fig. 10.

1) PFMEA justification and mapping: The baseline PFMEA ratings were calibrated against the Measure phase empirical findings (Table IX).

The resulting baseline PFMEA is presented in Table X.

The failure effect *"Specific windshield function disabled"* (e.g., ADAS camera bracket misalignment or acoustic layer absence) received $S = 7, O = 3, D = 7$, yielding RPN $= 147$. Although classified in the Moderate band ($100 < \text{RPN} \leq 150$) due to a lower historical frequency ($O = 3$), its high severity ($S = 7$) warrants direct mitigation. The automated machine vision system deployed in the Improve phase verifies complete homologation code strings, thereby safeguarding this failure mode simultaneously and reducing its detection score to $D = 1$.

E. Improve Phase

Corrective actions addressed the Method, Man, and Machine root causes (Table XI).

TABLE IV. LONGITUDINAL ANALYSIS OF PROCESS CAPABILITY, HIDDEN FACTORY, AND COPQ (2021–2025)

Year	Annual Volume (V)	External PPM (Escapes)	External Sigma (σ)	Sorting Volume (HF)	Internal PPM (Sorting)	Internal Sigma (σ)	Sorting Ratio (%)	Baseline COPQ Loss (EUR)
2021	1,000,000	12.00	5.72	9,600	9,600	3.84	0.96%	€57,600
2022	1,100,000	13.64	5.68	12,000	10,909	3.79	1.09%	€72,000
2023	1,400,000	17.14	5.63	19,200	13,714	3.71	1.37%	€115,200
2024	1,500,000	26.67	5.53	32,000	21,333	3.53	2.13%	€192,000
2025	1,600,000	29.38	5.52	37,600	23,500	3.49	2.35%	€225,600

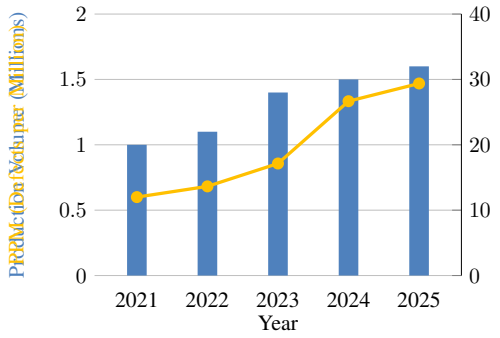


Fig. 6. Evolution of production volume vs PPM defect rate (2021–2025).

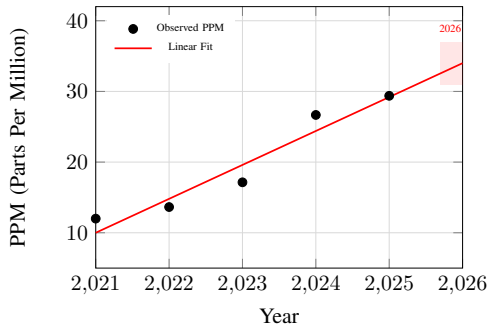


Fig. 7. PPM historical fit and descriptive projection ($R^2 = 0.94$).

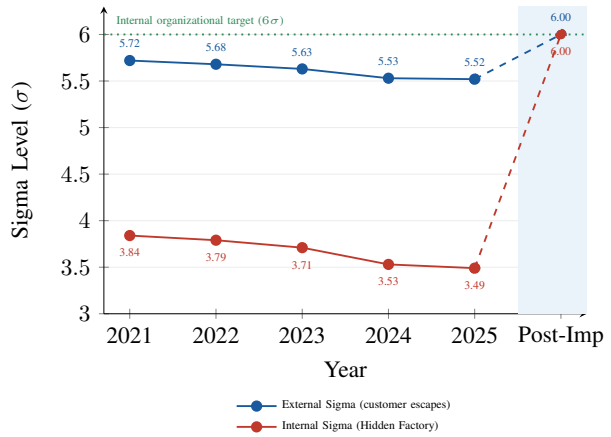


Fig. 8. Evolution of external and internal process capability (2021–2025) and convergence post-implementation.

TABLE V. ATTRIBUTE CONCORDANCE STUDY EXTRACT (FIRST 10 SAMPLES)

Sample	Standard Reference	Inspector 1	Inspector 2	Inspector 3	Concordance
1	Conforming (OK)	OK	OK	OK	Complete
2	Mix-up (NOK)	OK (error)	Mix-up	Mix-up	Disagree
3	Conforming (OK)	OK	OK	OK	Complete
4	Mix-up (NOK)	Mix-up	OK (error)	Mix-up	Disagree
5	Conforming (OK)	OK	Mix-up (error)	OK	Disagree
6	Conforming (OK)	OK	OK	OK	Complete
7	Conforming (OK)	OK	OK	OK	Complete
8	Mix-up (NOK)	Mix-up	Mix-up	Mix-up	Complete
9	Conforming (OK)	OK	Mix-up (error)	OK	Disagree
10	Conforming (OK)	OK	OK	OK	Complete

1) Technical architecture and optical configuration: The automated quality gate integrates three hardware components (Fig. 11):

TABLE VI. PER-SAMPLE AGREEMENT CALCULATIONS FOR EXTRACT SAMPLES

Sample	"OK" (n_{i1})	"Mix-up" (n_{i2})	Agreement (P_i)
1	3	0	$(3^2 + 0^2 - 3)/(3 \times 2) = 1.000$
2	1	2	$(1^2 + 2^2 - 3)/6 = 0.333$
3	3	0	1.000
4	1	2	0.333
5	2	1	0.333
6	3	0	1.000
7	3	0	1.000
8	0	3	1.000
9	2	1	0.333
10	3	0	1.000

TABLE VII. SUMMARY OF FLEISS' KAPPA PARAMETERS ($N = 150$, $n = 3$, $k = 2$)

Parameter	Symbol	Value
Number of samples	N	150
Number of inspectors	n	3
Number of rating categories	k	2
Mean observed agreement	$\bar{P}$	0.8267
Expected chance agreement	P_e	0.5525
Fleiss' Kappa	κ	**0.612**

TABLE VIII. LANDIS AND KOCH INTERPRETATION SCALE FOR CONCORDANCE [48]

κ Range	Strength of Agreement	Implication
< 0.00	Poor	Less than chance agreement
$0.00 - 0.20$	Slight	Minimal consistency
$0.21 - 0.40$	Fair	Some consistency
$0.41 - 0.60$	Moderate	Moderate consistency
0.61 – 0.80	**Substantial**	**Baseline study result** ($\kappa = 0.612$)
$0.81 - 1.00$	Almost Perfect	Near-complete agreement

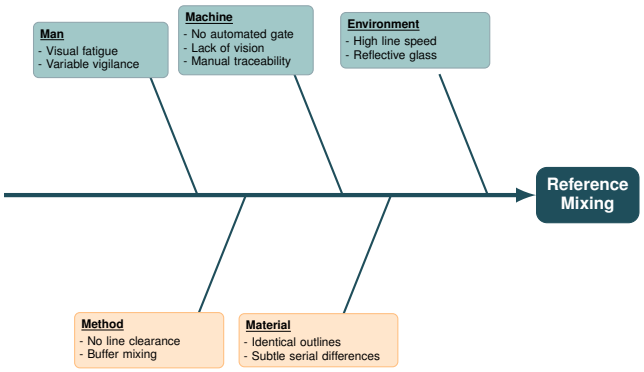


Fig. 9. Ishikawa diagram identifying root causes for reference mixing.

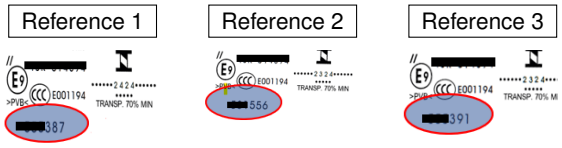


Fig. 10. Visual similarity among windshield references differing only in micro-printed regulatory codes.

TABLE IX. CALIBRATION OF PFMEA RATINGS AGAINST EMPIRICAL MEASURE-PHASE DATA

PFMEA Parameter	Empirical Input	Scale Mapping Justification
Severity ($S = 7$)	Cost and Customer Impact	Rating 7 denotes customer disruption (assembly line stoppage and functional mismatch) with €225,600 baseline losses.
Occurrence ($O = 4$)	Longitudinal Defect Rate	Under AIAG PFMEA criteria, $O = 4$ corresponds to an occurrence frequency of approximately 1 in 2,000 to 1 in 1,000 parts ($\sim$1,000 PPM batch risk during changeover shifts). With 47 escapes/year against active production runs, the baseline falls within rating 4.
Detection ($D = 7$)	Attribute Agreement Study	Rating 7 denotes low manual detection probability, corroborated by Fleiss' $\kappa = 0.612$.

- Vision Sensor: Cognex In-Sight 7802M monochrome camera (1600×1200 resolution, 1.3 MP) equipped with a 16 mm focal length lens. The working distance is set to 350 ± 15 mm, providing a field of view of 180×135 mm centered on the regulatory marking. Diffuse red LED ring lighting (630 nm) was selected to eliminate specular reflections from curved glass surfaces.
- Software Configuration: Programmed via Cognex In-Sight Explorer. Preprocessing applies localized contrast equalization. Pattern matching (Cognex PatMax) verifies logo presence and orientation with an acceptance threshold of 85%. Optical Character Recognition (Cognex OCRMax) segments and validates the homologation strings (ECE R43, DOT numbers) against master strings, with character confidence thresholds set at 80%.
- Control and Execution: A Siemens S7-1200 PLC interfaces with the camera over PROFINET. Image acquisition and classification require 400 ms within a total cell cycle time of 4.0 seconds. Downstream sorting is executed by an ABB IRB 6700 industrial robot.

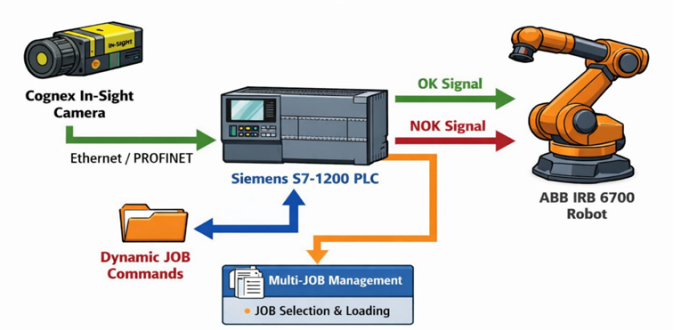


Fig. 11. System architecture integrating Cognex vision camera, Siemens S7-1200 PLC, and ABB sorting robot.

The physical layout on the shop floor is shown in Fig. 12.

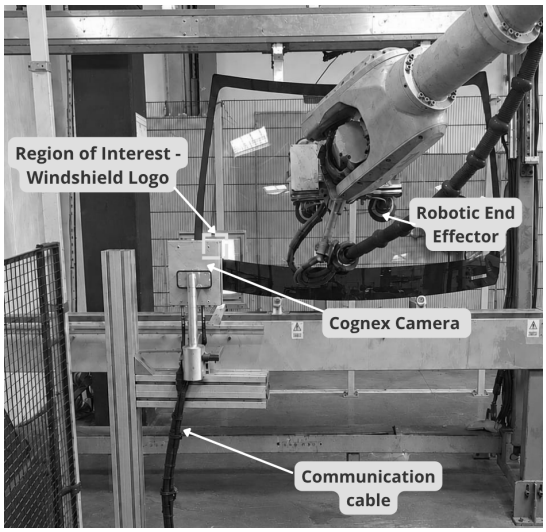


Fig. 12. Robotic presentation of windshield to the Cognex vision camera quality gate.

The operational sequence follows seven discrete steps:

- Reference program selection on HMI;
- Part loading;
- Robotic retrieval;
- Edge washing and drying;
- Manual visual check for surface scratches;
- Automated vision logo and reference inspection;
- Robotic sorting: conforming parts are released to packaging, while non-conforming parts are transferred to a quarantine rack (Fig. 13).

F. Control Phase

1) Validation protocols and statistical sample justification: A two-stage validation protocol was executed:

- Stage 1 (Pilot Challenge Test): A 30-day pilot test in January 2026 processed approximately 50,000 units

TABLE X. PFMEA EVALUATION FOR REFERENCE MIXING FAILURE MODES

Failure mode	Potential effect	S	Crit.	Potential cause	O	Current prevention	Current detection	D	RPN
Mixing reference	Customer assembly stoppage	7	F	Human fatigue / Visual similarity of logos / Lack of line clearance	4	None	100% manual visual inspection	7	196
	Vehicle mismatch	7	F		4	None		7	196
	Specific windshield function disabled	7	F		3	None		7	147

TABLE XI. ROOT CAUSES AND CORRESPONDING CORRECTIVE IMPLEMENTATIONS

Category	Root Cause Identified	Implemented Corrective Action
Method	Lack of changeover line clearance procedure.	Enforced line-clearance SOP: physical buffer purging prior to starting new production references.
Man	Insufficient risk awareness of visual similarity.	Operator training on micro-printed homologation differences and changeover discipline.
Machine	Absence of automated reference verification gate.	Deployment of a Cognex In-Sight machine vision cell with robotic rejection.

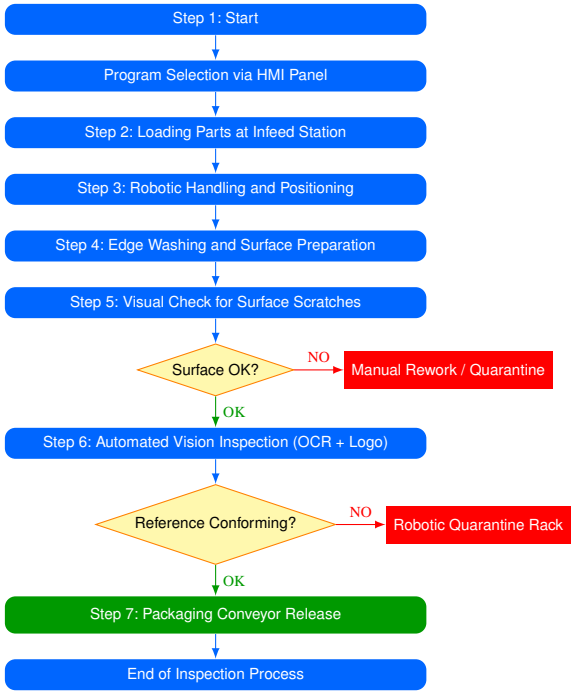


Fig. 13. Decision and sorting logic executed by the automated inspection cell.

across production shifts. A challenge set of 120 known non-conforming windshields (featuring mismatched homologation prints, alternative OEM logos, shifted stamps, and degraded ceramic ink) was systematically introduced across shifts. All 120 challenge units were successfully detected and rejected (100% sensitivity).

- Stage 2 (Audit Partition): A randomized audit set of $N_{\text{val}} = 300$ parts (including 50 seeded non-conforming units with subtle reference divergences) was evaluated. The resulting confusion matrix yielded:
  - True Negatives (TN): 246 conforming parts accepted.
  - True Positives (TP): 50 non-conforming parts rejected.
  - False Positives (FP): 4 conforming parts rejected (overkill = 1.6%).
  - False Negatives (FN): 0 non-conforming escapes (Type II error = 0.0%).

Based on this matrix, the performance metrics are:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{N_{\text{val}}} = \frac{50 + 246}{300} = 98.7\% \quad (12)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{50}{50 + 0} = 100.0\% \quad (13)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{246}{246 + 4} = 98.40\% \quad (14)$$

The attribute agreement statistic for this validation set yields $\kappa = 0.954$ (observed agreement $P_o = 0.9867$, chance agreement $P_e = 0.7133$), while extended Gage R&R multi-run trials confirmed overall attribute reliability at 0.98.

From a statistical standpoint, observing zero escapes in a sample of $N_{\text{val}} = 300$ yields a 95% one-sided Wilson confidence bound for the escape rate of $[0, 0.0099]$ (upper bound $\leq 0.99\%$). This validation sample confirms zero Type II error under test conditions. However, verifying compliance at extreme quality targets (6σ / 3.4 PPM) cannot be proven by a 300-unit sample alone. Operational proof was therefore established through the subsequent six-month longitudinal monitoring of 700,000 units.

2) Longitudinal monitoring and process isolation: Over six consecutive months, the automated cell inspected the full line throughput of 700,000 windshields. During this period, zero customer escapes were recorded. The vision system triggered 345 total alarms:

- True Positives (35 units): Exactly 5 units were confirmed reference mix-ups, and 30 units exhibited severe ceramic print defects or illegible homologation characters.
- False Positives / Overkill (310 units): Conforming units flagged due to surface reflections or low ink contrast (overkill rate = 0.044%).

This outcome allows the effects of procedural controls to be isolated from the machine vision gate. While line-clearance standards and training reduced upstream mix-up frequency, 5 actual mix-ups still occurred during changeovers. Without the

automated vision gate, these 5 units would have escaped to customers. The automated gate thus served as the necessary containment barrier.

3) Updated key performance indicators and RPN: Table XII provides a comparison between the baseline and the control phase.

TABLE XII. COMPARATIVE ANALYSIS OF BASELINE VERSUS AUTOMATED CONTROL STATE

Performance Metric	Manual Baseline (2025)	Control Phase (Vision)	Improvement
RPN ($S \times O \times D$)	196 ($7 \times 4 \times 7$)	14 ($7 \times 2 \times 1$)	-92.9%
Customer Escapes	47 cases/year	0 cases (in 6-month run)	-100.0%
Attribute Measurement Reliability	$\kappa = 0.612$ (Human)	$\kappa = 0.954$ (Validation set)	$+55.9\%$
Classification Accuracy	$\sim 75.0\%$ (estimated)	98.7% (Sens: 100%, Spec: 98.4%)	$+23.7\%$
Baseline COPQ Incurred	€225,600 / year	€0 / year (escapes eliminated)	-100.0%
Inspection Cycle Time	5.0 s	4.0 s	-20.0%

The RPN decreased from 196 to 14 (Fig. 14). Severity remained at $S = 7$. Occurrence was reduced to $O = 2$ due to line-clearance SOPs, and Detection was updated to $D = 1$ based on 100% in-line robotic gatekeeping.

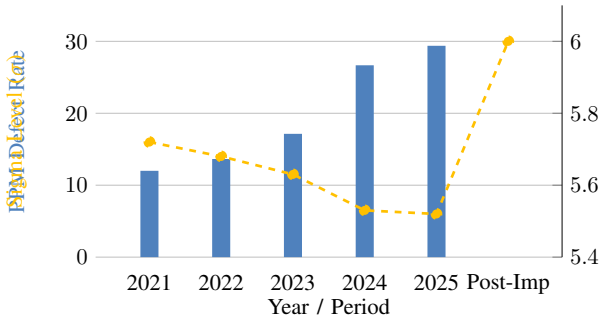


Fig. 14. Annual PPM and Sigma capability trends through post-implementation.

The human-machine interface (HMI) logs every inspection event, displays live counts, and manages alarm states (Fig. 15). Process sustainability is reinforced through a daily Master Piece ("Red Piece") protocol: at shift handover, a known non-conforming part is presented to verify automatic rejection before production is unlocked.

V. DISCUSSION

A. Human Vigilance Versus Engineered Poka-Yoke Gates

The shift in attribute agreement from $\kappa = 0.612$ to $\kappa = 0.954$ confirms that manual visual discrimination of micro-printed characters at line cadence is fundamentally limited by cognitive fatigue. Reducing the Detection score from $D = 7$ to $D = 1$ substituted a variable human task with an automated Poka-Yoke gate [49]. During the six-month production run, the system intercepted 5 true reference mix-ups that had breached upstream procedural clearance, demonstrating that procedural discipline alone is insufficient in high-mix facilities.

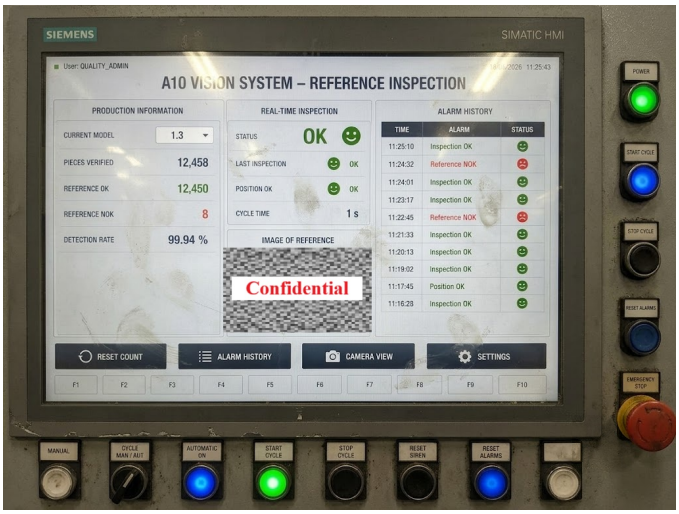


Fig. 15. Line-side HMI dashboard displaying real-time inspection counts, defect statistics, and alarm histories.

B. Economic Analysis and Comprehensive Return on Investment

Historical baseline COPQ losses were €225,600 in 2025. Accounting for operating expenses (OPEX)—including annual optics maintenance (€2,500), bi-monthly calibration checks (€1,200), and false-alarm technician review time (14 cumulative downtime hours over six months, valued at €3,360)—the net annual financial recovery is calculated as:

$$\begin{aligned} \text{Net Annual Recovery} &= €225{,}600 - €7{,}060 \\ &= €218{,}540/\text{year} \end{aligned} \tag{15}$$

Because the industrial robot and PLC were existing assets, the initial capital expenditure (CAPEX) was limited to the Cognex camera, lens, diffuse illumination, and software integration (€18,000). The resulting payback period was achieved in approximately one month:

$$\text{Payback Period} = \left(\frac{€18{,}000}{€218{,}540/12} \right) \approx 0.99 \text{ months} \tag{16}$$

C. Trade-Off Between Sensitivity and Overkill

The observed overkill rate of 0.044% (310 false rejections out of 700,000 parts) reflects a deliberate safety-first configuration. In safety-critical automotive glazing, false rejections routed to technician review carry minimal operational penalties relative to the catastrophic warranty and containment costs of an escape.

D. Technical Limitations

Three operational limitations are recognized:

1) Program management: Introducing new windshield references requires generating new region-of-interest templates and syntax strings, adding engineering setup time [50].

2) Optical reflection: Residual curvature reflections can occasionally cause false rejections on high-gloss coatings [52], [51].

The single-plant case design means financial and cycle time gains depend on the specific line layout and existing automation infrastructure.

E. Secondary Quality Capabilities

An unexpected benefit of the deployment was the incidental detection of ceramic screen-printing anomalies. Faded ceramic ink, character clipping, and logo smudging caused the OCR engine to fail string verification, flagging 30 marking defects and functioning as a secondary print-quality gate without additional capital expenditure [53].

VI. Conclusion

This study demonstrated the structured application of the DMAIC framework to eliminate reference-mixing defects in automotive windshield manufacturing. By deploying an automated Poka-Yoke quality gate integrating a Cognex machine vision camera, Siemens PLC, and ABB sorting robot, reference escapes were reduced from 47 incidents per year to zero across a six-month monitoring run of 700,000 windshields.

Attribute measurement reliability improved from $\kappa = 0.612$ to $\kappa = 0.954$, and the baseline RPN decreased from 196 to 14. Net annual COPQ savings reached €218,540, recovering the initial €18,000 capital investment in under one month. Future work will investigate integrating deep learning-based OCR architectures to further suppress false rejections on low-contrast ceramic prints while maintaining absolute defect containment.

References

[1] M. Ronchi, M. Gamberi, C. Cafarella, A. Caporale, F. G. Galizia, et al., "Automatic visual inspection in automotive assembly lines: a techno-economic assessment for the configuration check of exterior components," in *Proc. Summer School Francesco Turco*, 2024, pp. 1–6.

[2] V. Gurav and A. Upadhyay, "Quality assurance in mixed-model assembly lines through vision-based deep learning," *Industrial Engineering Journal*, vol. 54, no. 1, 2025, doi: 10.13140/RG.2.2.19639.64161.

[3] C. A. de Souza Silva and E. P. Paladini, "Smart machine vision system to improve decision-making on the assembly line," *Machines*, vol. 13, no. 2, p. 98, 2025.

[4] H.-D. Lin, H.-L. Wu, and C.-H. Lin, "A deep transfer learning-based visual inspection system for assembly defects in similar types of manual tool products," *Sensors*, vol. 25, no. 6, p. 1645, 2025.

[5] R. Y. Alsada and Y. Kumar, "A measurement of quality costs in industrial organizations," *Cogent Business & Management*, vol. 9, no. 1, p. 2128253, 2022.

[6] C. Ashley and S. Varki, "Loyalty and its influence on complaining behavior and service recovery satisfaction," *Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior*, vol. 22, pp. 21–35, 2009.

[7] N. Ngoc Quang and D. C. Thuy, "Justice and trustworthiness factors affecting customer loyalty with mediating role of satisfaction with complaint handling: Zalo OTT Vietnamese customer case," *Cogent Business & Management*, vol. 10, no. 2, p. 2211821, 2023.

[8] A. Rocha, A. Pedregal, C. Maia, G. Soares, H. Cardoso, and M. Fonseca, "Complaints management: quality of response to customer complaints in an automotive company," *International Journal of Productivity and Quality Management*, vol. 44, no. 1, pp. 18–39, 2025.

[9] M. Martin, X. Centelles, A. Sole, C. Barreneche, A. I. Fernández, and L. F. Cabeza, "Polymeric interlayer materials for laminated glass: A review," *Construction and Building Materials*, vol. 230, p. 116897, 2020.

[10] A. Mareška, T. Kordová, M. Havlík Míka, et al., "Relaxation of polyvinyl butyral film with non-uniform thickness," *Manufacturing Technology*, vol. 22, no. 1, pp. 45–58, 2022.

[11] S. B. Jha and R. F. Babiceanu, "Deep CNN-based visual defect detection: Survey of current literature," *Computers in Industry*, vol. 148, p. 103911, 2023.

[12] H. A. Khorshidi, I. Gunawan, and F. Esmaeilzadeh, "Implementation of SPC with FMEA in less-developed industries with a case study in car battery manufactory," *International Journal of Quality and Innovation*, vol. 2, no. 2, pp. 148–157, 2013.

[13] F. Sumasto, Y. A. Nugroho, E. S. Solih, A. W. Arohman, D. Agustin, and A. K. Permana, "Enhancing quality control in the Indonesian automotive parts industry: A defect reduction approach through the integration of FMEA and MSA," *Instrumentation, Mesures, Métrologies*, vol. 23, no. 1, 2024.

[14] R. Godina, J. Matias, and S. Azevedo, "Quality improvement with statistical process control in the automotive industry," *International Journal of Industrial Engineering and Management*, vol. 7, no. 1, p. 1, 2016.

[15] R. Godina, C. Pimentel, F. Silva, and J. C. Matias, "Improvement of the statistical process control certainty in an automotive manufacturing unit," *Procedia Manufacturing*, vol. 17, pp. 729–736, 2018.

[16] L. Cepova, A. Kovacikova, R. Cep, P. Klaput, and O. Mizera, "Measurement system analyses–Gauge repeatability and reproducibility methods," *Measurement Science Review*, vol. 18, no. 1, pp. 20–27, 2018.

[17] I. A. Memon, Q. B. Jamali, A. S. Jamali, M. K. Abbasi, N. A. Jamali, and Z. H. Jamali, "Defect reduction with the use of seven quality control tools for productivity improvement at an automobile company," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 4044–4047, 2019.

[18] J. Doshi and D. Desai, "Application of failure mode & effect analysis (FMEA) for continuous quality improvement-multiple case studies in automobile SMEs," *International Journal for Quality Research*, vol. 11, no. 2, p. 345, 2017.

[19] M. Soković, D. Pavletić, and E. Krulčić, "Six Sigma process improvements in automotive parts production," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 19, no. 1, pp. 96–102, 2006.

[20] N. Vamsi Krishna Jasti and A. Sharma, "Lean manufacturing implementation using value stream mapping as a tool: A case study from auto components industry," *International Journal of Lean Six Sigma*, vol. 5, no. 1, pp. 89–116, 2014.

[21] A. M. Reis, A. Dall-Orsoletta, E. Nunes, L. Costa, and S. Sousa, "Quality costs and Industry 4.0: inspection strategy modelling and reviewing," *The International Journal of Advanced Manufacturing Technology*, vol. 136, no. 9, pp. 3883–3897, 2025.

[22] K. Knop, E. Olejarz, and R. Ulewicz, "Evaluating and improving the effectiveness of visual inspection of products from the automotive industry," in *Int. Scientific-Technical Conf. MANUFACTURING*, Springer, 2019, pp. 231–243.

[23] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao, "Surface defect detection methods for industrial products: A review," *Applied Sciences*, vol. 11, no. 16, p. 7657, 2021.

[24] E. Indasyah, F. Ibrahim, D. F. Syahbana, and F. Istiqomah, "Automated visual inspection system of gear surface defects detection using faster RCNN," in *2023 Int. Conf. Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, IEEE, 2023, pp. 899–904.

[25] A. Pacana, K. Czerwińska, and R. Dwornicka, "Analysis of quality control efficiency in the automotive industry," *Transportation Research Procedia*, vol. 55, pp. 691–698, 2021.

[26] J. Buckley and R. Servent, "Improvements in ultrasonic inspection of resistance spot welds," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 51, no. 2, pp. 73–77, 2009.

[27] KEYENCE America, "Glass inspection for the automotive industry," https://www.keyence.com, 2024.

[28] J. Carroll, "Machine vision system detects antenna trace defects on automotive glass," *Vision Systems Design*, vol. 24, p. 16, 2019.

[29] P. Ding and L. Yang, "Glass defect detection with improved data augmentation under total reflection lighting," *Applied Sciences*, vol. 14, no. 13, p. 5658, 2024.

[30] Z. Chen, S. Huang, H. Lv, Z. Luo, and J. Liu, "Defect detection in automotive glass based on modified YOLOv5 with multi-scale feature fusion and dual lightweight strategy," *The Visual Computer*, vol. 40, no. 11, pp. 8099–8112, 2024.

[31] S. Lee, B. Kim, and H. Lee, "Deep learning-based glass detection for smart glass manufacturing processes," *Computers, Materials and Continua*, vol. 84, no. 1, pp. 1397–1415, 2025.

[32] F. Sari and A. B. Ulas, "Deep learning application in detecting glass defects with color space conversion and adaptive histogram equalization," *Traitement du Signal*, vol. 39, no. 2, p. 731, 2022.

[33] C.-C. Ho, H.-P. Wang, and Y.-C. Chiao, "Machine vision-based cutting process for LCD glass defect detection system," *The International Journal of Advanced Manufacturing Technology*, vol. 123, no. 5, pp. 1477–1498, 2022.

[34] A. El Mazgualdi, H. Masrour, A. El Hassani, and J. El Mhamdi, "Visual inspection and scratch detection on glass panels using convolutional neural networks," *Journal of Glass Science and Technology*, vol. 42, no. 3, pp. 115–124, 2023.

[35] E. Adán and A. Adán, "Computer vision for glass waste: Technologies and sensors," *Sensors*, vol. 25, no. 21, p. 6634, 2025.

[36] T. Yamaura, K. Kobayashi, and M. Watanabe, "Depth estimation of internal latent flaws in glass substrates using light-scattering imaging and machine learning," *Optical Engineering*, vol. 64, no. 2, p. 023101, 2025.

[37] Q. Zhou, R. Chen, B. Huang, C. Liu, J. Yu, and X. Yu, "An automatic surface defect inspection system for automobiles using machine vision methods," *Sensors*, vol. 19, no. 3, p. 644, 2019.

[38] N. Hütten, R. Meyes, and T. Meisen, "Vision transformer in industrial visual inspection," *Applied Sciences*, vol. 12, no. 23, p. 11981, 2022.

[39] Y.-T. Jou, R. M. Silitonga, M.-C. Lin, R. Sukwadi, and J. Rivaldo, "Application of Six Sigma methodology in an automotive manufacturing company: a case study," *Sustainability*, vol. 14, no. 21, p. 14497, 2022.

[40] L. M. Monday, "Define, measure, analyze, improve, control (DMAIC) methodology as a roadmap in quality improvement," *Global Journal on Quality and Safety in Healthcare*, vol. 5, no. 2, pp. 44–46, 2022.

[41] V. Gupta, R. Jain, M. Meena, and G. Dangayach, "Six-sigma application in tire-manufacturing company: a case study," *Journal of Industrial Engineering International*, vol. 14, no. 3, pp. 511–520, 2018.

[42] M. Bhargava and S. Gaur, "Process improvement using six-sigma (DMAIC process) in bearing manufacturing industry: a case study," in *IOP Conf. Series: Materials Science and Engineering*, vol. 1017, 2021, p. 012034.

[43] R. G. Schroeder, K. Linderman, C. Liedtke, and A. S. Choo, "Six Sigma: Definition and underlying theory," *Journal of Operations Management*, vol. 26, no. 4, pp. 536–554, 2008.

[44] Y. H. Kwak and F. T. Anbari, "Benefits, obstacles, and future of six sigma approach," *Technovation*, vol. 26, no. 5-6, pp. 708–715, 2006.

[45] V. Grecu and C. Deneş, "The hidden factory: Quantifying informational waste and operational capacity loss in manufacturing," *Acta Universitatis Cibiniensis. Technical Series*, vol. 77, no. 1, pp. 55–64, 2025.

[46] H. Rødseth, P. Schjølberg, and A. Marhaug, "Deep digital maintenance," *Advances in Manufacturing*, vol. 5, no. 4, pp. 299–310, 2017.

[47] V. Tuominen, "Cost modeling of inspection strategies in automotive quality control," *Engineering Management Research*, vol. 1, no. 2, p. 33, 2012.

[48] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia Medica*, vol. 22, no. 3, pp. 276–282, 2012.

[49] D. Romero, P. Gaiardelli, D. J. Powell, and M. Zanchi, "Intelligent poka-yokes: error-proofing and continuous improvement in the digital lean manufacturing world," in *IFIP Int. Conf. Advances in Production Management Systems*, Springer, 2022, pp. 595–603.

[50] N. Alsaadi, "Roadblocks in integrating Lean Six Sigma and Industry 4.0 in small and medium enterprises," *Systems*, vol. 12, no. 4, p. 125, 2024.

[51] M. Maślanka, D. Jancarczyk, and J. Rysinski, "Integration of machine vision and PLC-based control for scalable quality inspection in Industry 4.0," *Sensors*, vol. 25, no. 20, p. 6383, 2025.

[52] M. Hussain, "Sustainable machine vision for Industry 4.0: a comprehensive review of convolutional neural networks and hardware accelerators in computer vision," *AI*, vol. 5, no. 3, 2024.

[53] O. Morales Matamoros, J. G. Takeo Nava, J. J. Moreno Escobar, and B. A. Ceballos Chávez, "Artificial intelligence for quality defects in the automotive industry: A systemic review," *Sensors*, vol. 25, no. 5, p. 1288, 2025.

Hodge-Guided Active Preference Elicitation for Efficient Pairwise Ranking

Eskander Bejaoui*, Mohamed Ould-Elhassen Aoueileyine, Ridha Bouallegue
Innov'COM Laboratory, Higher School of Communication of Tunis, University of Carthage, Tunis, Tunisia

***Abstract*—Pairwise comparison is a standard method for eliciting user preferences in recommender systems and group decision-making. The number of required comparisons grows quadratically with the number of alternatives, creating a substantial burden for users and platforms. This paper asks whether Hodge decomposition, which splits a preference flow into gradient (consensus), curl (local inconsistency), and harmonic (global inconsistency) components, can guide which comparisons to request, rather than merely aggregate whichever comparisons arrive. Preference elicitation is formalized as a sequential acquisition problem for online HodgeRank. Seven acquisition policies are evaluated, including two novel Hodge-informed policies: HarmonicReduce, which targets edges likely to carry harmonic inconsistency, and FlowClone, a lightweight neural policy trained by behavioural cloning on a compact five-dimensional feature representation. A Bayesian Bradley-Terry active-learning baseline (BALD) is also included. Policies are evaluated on three real-world datasets under a leakage-free protocol: held-out relevance labels come from full-data batch HodgeRank, tracked via normalized discounted cumulative gain, and validated under five-fold leave-users-out cross-validation so every user is evaluated once. Across all three datasets, geometry-informed active policies cut the comparisons needed to reach a fixed recommendation-quality threshold by roughly half relative to random sampling (43–56% depending on dataset). They consistently outperform BALD, which is itself a statistically confirmed improvement over random sampling on two of the three datasets. This advantage depends on the scoring model: pairing active acquisition with a Borda count aggregator produces worse outcomes than random acquisition with HodgeRank, a phenomenon termed the Borda collapse and confirmed significant, in both directions, by paired Wilcoxon tests with Holm-Bonferroni correction. HarmonicReduce's effectiveness depends on comparison-graph density, a pattern proven exactly from the circuit rank of the comparison graph. These findings show that Hodge-theoretic structure is informative not only for aggregating collected preferences, but also for deciding which preferences to collect.**

***Keywords*—HodgeRank; hodge decomposition; active learning; preference elicitation; pairwise comparisons; uncertainty sampling; behavioural cloning; recommender systems**

I. Introduction

Pairwise comparison is a default mechanism for collecting subjective human judgments. It is used in recommender systems, crowdsourced quality assessment, and the reward-model training pipelines behind reinforcement learning from human feedback (RLHF) for large language models [1], [2], [3]. Compared with direct numerical scoring, pairwise comparisons avoid inconsistent scale interpretation across raters and are cognitively simpler to produce reliably [4]. This robustness has a combinatorial cost: a complete comparison set over n alternatives requires $\binom{n}{2}$ judgments, growing quadratically as the item catalogue grows. Platforms therefore face a direct trade-off between recommendation quality and user burden. Too few comparisons yield an unstable ranking. Too many risk user fatigue, attrition, or abandonment before elicitation completes, especially during cold-start onboarding when no recommendation can yet be shown. The same cost-of-comparisons tension appears at a different scale in RLHF pipelines, which are explicitly constrained by the cost of human-annotated preference pairs, motivating a growing literature on which comparisons a system should request rather than passively accept [5], [6], [7], [8]. This motivates a direct question: rather than treating the arrival order of comparisons as fixed or random, can a system choose, at each step, the comparison that most reduces its uncertainty about the underlying ranking, and thereby reach an equally reliable recommendation with substantially fewer comparisons?

HodgeRank is an algebraic ranking method built on combinatorial Hodge theory. It decomposes a pairwise-comparison flow uniquely and orthogonally into a gradient component (a globally consistent ranking), a curl component (local, triangle-level inconsistency), and a harmonic component (global inconsistency that no local triangle can detect) [9], [10]. Prior work extends this decomposition from an aggregate consistency diagnostic to per-contributor behavioural signatures, deriving curl- and harmonic-energy-based features to weight or assess contributors after their comparisons are collected [11]. What remains open is whether the same orthogonal split, into consensus, local inconsistency, and global inconsistency, is informative for acquisition: deciding which comparison to request next, rather than how to weight, aggregate, or audit comparisons already collected. This is a natural question. If curl and harmonic energy identify which existing comparisons are inconsistent, the same quantities, computed incrementally as new comparisons arrive, might identify which future comparisons are most likely to resolve inconsistency, and therefore most valuable to request.

Active learning formalizes a simple intuition: a learner that chooses its own training examples can reach a target accuracy with far fewer labels than one that receives them passively [12]. For preference data specifically, Bayesian approaches select comparisons that maximize expected information gain over a Gaussian-process preference model [13], [14]. Query-complexity results show that an adaptively chosen sequence of pairwise comparisons can recover a ranking using far fewer queries than random or exhaustive comparison, when items admit a low-dimensional embedding [15]. The recent growth

*Corresponding author.

of RLHF has renewed interest in this problem under the name query-efficient preference collection, typically operating on a scalar Bradley–Terry reward model [5], [7], [6], [8]. Despite this convergence of interest, no prior work asks whether Hodge-theoretic structure specifically is informative for acquisition, rather than aggregation or quality assessment.

This paper makes four contributions. First, sequential preference elicitation for online HodgeRank is formalized under a leakage-free evaluation protocol: held-out relevance labels come from a full-data batch ranking, not from the online policy's own trajectory. The learned policy is validated under five-fold leave-users-out cross-validation, so every user is evaluated exactly once. Second, two novel Hodge-informed acquisition policies are introduced. HarmonicReduce targets pairs likely to carry harmonic energy via a degree-weighted residual proxy. FlowClone is a lightweight neural network trained by behavioural cloning to approximate an uncertainty-sampling expert from a compact five-dimensional feature vector. A Bayesian Bradley-Terry active-learning baseline (BALD) is additionally implemented as a concrete comparison against the Bayesian preference-elicitation literature. Third, seven acquisition policies are evaluated against four scoring models on three real-world datasets of differing size and density, documenting a structural incompatibility, the Borda collapse, between gradient-residual acquisition and Borda-count scoring. Fourth, a combinatorial proposition based on the circuit rank of the comparison graph is proven. It explains exactly why HarmonicReduce's effectiveness switches sign between dense and sparse comparison graphs.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the mathematical framework. Section IV describes the experimental setup. Section V reports results. Section VI discusses implications and limitations. Section VII concludes.

II. Related Work

Aggregating pairwise comparisons into a global ranking is classically addressed by Borda count [4], Copeland's method, and the Bradley–Terry probabilistic model [16]. The author in [9] introduced combinatorial Hodge theory to this problem, formalizing the orthogonal gradient/curl/harmonic decomposition used throughout this paper. [10] applied it to subjective video quality assessment. Follow-up work extends the decomposition to per-contributor behavioural analysis, using curl and harmonic energy to weight or assess contributors after the fact, rather than purely for aggregate consistency assessment [11]. All of this prior work, including acquisition-agnostic evaluations of HodgeRank's stability under sparse, randomly sampled data [17], treats the comparison set as given and fixed. None asks whether the same decomposition can inform which comparison to request next. This paper is, to our knowledge, the first to use Hodge-theoretic structure prospectively, to decide which comparison to request next, rather than retrospectively, to weight, aggregate, or audit comparisons already collected.

Active learning rests on a simple observation: a learner able to select its own training instances can reach a target performance with substantially fewer labels than passive sampling [12]. For preference data, [13] model preferences with a Gaussian process and select comparisons by expected information gain. [14] generalize this to Bayesian active learning (BALD). These approaches typically require a probabilistic model with a tractable posterior over item utilities. HodgeRank, a deterministic weighted least-squares estimator, does not directly provide one. Our Entropy policy (Section III-C) is a deterministic analogue of uncertainty sampling that uses the current model's residual as an informativeness proxy. We additionally implement a genuine Bayesian baseline, an online Bayesian Bradley-Terry model with a Gaussian-probit moment update adapted from [18] for pairwise outcomes, combined with BALD-style acquisition, to give an empirical rather than purely narrative comparison. In the active-ranking theory literature, [15] show that if items admit a low-dimensional Euclidean embedding, an adaptive algorithm can recover a ranking using far fewer comparisons than passive sampling. HodgeRank's gradient component is precisely such a low-dimensional embedding, giving a theoretical anchor for the empirical query-complexity reduction reported in Section V.

Recent literature studies active or query-efficient preference collection motivated by RLHF annotation cost. This includes acquisition strategies for direct preference optimization [5], [3], provable query-complexity RLHF algorithms under a Bradley–Terry reward model [7], sample-efficient active preference optimization [6], and experimental-design criteria balancing exploration against informativeness [8]. These works share our motivation: preference labels are expensive and acquisition order matters. But they operate on a scalar Bradley–Terry-style reward model rather than the geometric, decomposition-based representation used here. Our scorer ablation (Section V-C) directly probes this distinction. Finally, behavioural cloning trains a policy by supervised imitation of an expert's choices rather than by reward maximization. It remains an efficient alternative to reinforcement learning when the reward signal is sparse. We adopt it for FlowClone because a policy directly trained on rewards, optimized against per-step nDCG improvement, suffers from high-variance, sparse-reward updates.

III. Methodology

A. Problem Formulation

The acquisition question this paper asks, which comparison should be requested next, is meaningful only relative to a single respondent's own trajectory of answers. We therefore frame everything at the level of one user u, not the full population, and defer aggregation across users to the evaluation protocol (Section III-E). Fix an item set $X = \{x_1, \ldots, x_n\}$ for that user. Comparisons arrive one at a time as ordered pairs (w, ℓ), read as "w was chosen over ℓ." We reserve 20% of each user's available comparisons, denoted $P^{(u)}_{\text{held}}$, purely as an evaluation target never exposed to any policy. This leaves a train pool $P^{(u)}_{\text{train}}$ from which an acquisition policy π sequentially draws. At each step t, π picks a still-unqueried pair (w_t, ℓ_t) from $P^{(u)}_{\text{train}}$, receives the outcome, and folds it into a running per-user score estimate. This estimate is updated after every single comparison, not recomputed in batch.

The candidate action space at step t is therefore $P^{(u)}_{\text{train}} \setminus Q_t$, the still-unqueried comparisons that already appear with their observed winner in that user's historical record, not the full $\binom{n}{2}$ set of possible item pairs. This is an offline-replay

setting: a policy cannot request a pair the user never actually compared, and it cannot observe an outcome other than the one the user originally gave. Offline replay approximates genuine online elicitation only to the extent the original data-collection process is representative of how the same user would answer if queried live, in the same order a policy proposes. None of the used datasets carries contextual or presentation-order effects (e.g., item position, screen layout, or fatigue over the course of a session) because the source data records only comparison outcomes, not the context under which each comparison was made; a genuinely live deployment could exhibit such effects, and offline replay on either dataset is silent on them. This scopes offline-replay validity to the acquisition-order question this paper asks.

B. HodgeRank and Hodge Decomposition

We adopt the standard HodgeRank estimator to obtain a reference (non-sequential) score for a user once all of that user's comparisons are in hand. Write $c_{ij}^{(u)}$ for how many times user u chose item i over item j. Each compared pair is summarized by a normalized margin $\hat{Y}_{ij}^{(u)} = (c_{ij}^{(u)} - c_{ji}^{(u)})/(c_{ij}^{(u)} + c_{ji}^{(u)}) \in [-1, 1]$ and a confidence weight $\omega_{ij}^{(u)} = c_{ij}^{(u)} + c_{ji}^{(u)}$, reflecting how often that pair was tested. The batch score vector $s^{(u)} \in \mathbb{R}^n$ is the vector that best reproduces these margins as score differences, in a weighted least-squares sense:

$$\min_{s^{(u)}} \sum_{\{i,j\}} \omega_{ij}^{(u)} \big(s_i^{(u)} - s_j^{(u)} - \hat{Y}_{ij}^{(u)}\big)^2 + \lambda \|s^{(u)}\|^2, \qquad (1)$$

regularized by $\lambda = 10^{-5}$ and solved with sparse LSQR [19]. This estimator is useful beyond curve-fitting because its residual is not arbitrary noise. The observed margin vector y splits uniquely into three pairwise-orthogonal pieces, $y = y^g + y^c + y^h$. The first, $y_{ij}^g = s_i^* - s_j^*$, is exactly what the fitted score vector predicts: the part of the data consistent with a single global ranking. The remainder splits further into a piece explainable by local disagreement confined to individual triangles of items (y^c), and a piece that cannot be pinned to any single triangle at all (y^h). This harmonic piece is simultaneously curl-free and divergence-free, and reflects inconsistency distributed around longer cycles in the comparison graph. It is the piece this paper cares about most for acquisition: it vanishes identically whenever the comparison graph has no cycles, and only becomes possible once enough comparisons close a loop.

For elicitation, we cannot wait for a full comparison set before producing a score. We instead maintain $s^{(u)}$ online, updating it after each single comparison rather than solving Eq. (1) from scratch. With warm-up constant γ_0, step index t, and a $1/(t + \gamma_0)$ learning-rate schedule, each newly observed pair (w, ℓ) triggers:

$$\eta_t = \frac{1}{t + \gamma_0}, \quad r_t = s_w^{(u,t)} - s_\ell^{(u,t)} - 1, \quad s_{w,\ell}^{(u,t+1)} = s_{w,\ell}^{(u,t)} \mp \eta_t r_t, \qquad (2)$$

The residual r_t plays a dual role. It drives the score update above, and it also serves as a live signal for how surprising the just-observed comparison was, a signal several acquisition policies below repurpose as a proxy for informativeness. γ_0 is set per dataset to one-third of that dataset's median per-user comparison count.

Eq. (2) is a stochastic-approximation scheme, in the sense of Robbins and Monro [20], for the same objective Eq. (1) minimizes in the unregularized limit: each step moves $s_w^{(u,t)}$ and $s_\ell^{(u,t)}$ along the negative gradient of a single comparison's squared-residual term, scaled by a learning rate $\eta_t = 1/(t+\gamma_0)$ satisfying the standard Robbins–Monro conditions $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$ once t is large enough. It is therefore an approximation that converges toward the batch minimizer as more comparisons accumulate, not an independent heuristic unrelated to Eq. (1), though at any finite t, and especially early in a trajectory, its running estimate need not agree with the batch solution over the same comparisons. Section V-G verifies this relationship empirically.

C. Acquisition Policies

Let $c_i^{(t)}$ be the number of times item i has appeared in a query so far and $\mathcal{N}_i^{(t)}$ its already-compared neighbours. We evaluate seven policies:

Random selects uniformly from the remaining pool, and serves as the baseline.

Popularity-LB concentrates queries on already-popular items, $\arg\max_{(i,j)}(c_i^{(t)} + c_j^{(t)})$, a simple heuristic baseline distinct from the geometry-informed policies.

Entropy is a deterministic analogue of uncertainty sampling, selecting the pair with the largest current residual, $\arg\max_{(i,j)} |s_i^{(t)} - s_j^{(t)} - 1|$.

Coverage expands the comparison graph structurally, preferring pairs with few shared already-queried neighbours, with the residual used only to break ties.

HarmonicReduce, a novel policy, targets comparisons likely to carry harmonic energy via a query-count-weighted residual proxy,

$$\pi_{\text{hr}}(t) = \arg\max_{(i,j)} \; |r_{ij}^{(t)}| \cdot \log\left(1 + \tfrac{c_i^{(t)} + c_j^{(t)} + 2}{2}\right). \qquad (3)$$

The harmonic subspace of a graph has dimension exactly its circuit rank $\beta_1 = |E| - |X| + c$, where c is the number of connected components; harmonic energy is nonzero only when $\beta_1 > 0$ [9], [21].

Proposition 1. *Let $G_t = (X, E_t)$ be the comparison graph after t steps and $(i, j) \notin E_t$ a candidate edge. Adding (i, j) increases β_1 by exactly 1 if i, j already lie in the same connected component, and by 0 otherwise.*

Proof: Adding an edge changes $|E|$ by $+1$. If i, j are already connected, c is unchanged, so $\beta_1 = |E| - |X| + c$ increases by 1. If they lie in different components, the edge merges them, so c decreases by 1 and β_1 is unchanged. ■

An edge that merely extends the spanning forest cannot carry harmonic energy; only an edge that closes

a cycle can. Query-count $c_i^{(t)}$ is a cheap proxy for $\Pr[i, j \text{ already connected}]$. Eq. (3)'s degree weighting targets exactly the precondition Proposition 1 identifies. On a sparse graph this proxy is discriminative, since most pairs remain unconnected for much of the elicitation budget. On a dense, near-complete graph, almost every pair is already connected after a handful of queries, so the proxy degenerates to a near-constant multiplier. This explains HarmonicReduce's density-dependent performance in Section V-D.

FlowClone is a 5-64-64-1 multilayer perceptron f_θ with $\tanh$ activations, scoring candidate pairs from a compact feature vector

$$\mathbf{x}_{ij}^{(t)} = [\rho_{ij}^{(t)},\, \sigma_{ij}^{(t)},\, \delta_{ij}^{(t)},\, \kappa_{ij}^{(t)},\, \tau_t]^\top, \tag{4}$$

where,

$\rho_{ij}^{(t)} = |r_{ij}^{(t)}|, \sigma_{ij}^{(t)} = s_i^{(t)} + s_j^{(t)}, \delta_{ij}^{(t)} = |s_i^{(t)} - s_j^{(t)}|, \kappa_{ij}^{(t)} = \log\big(1 + |\mathcal{N}_i^{(t)} \cap \mathcal{N}_j^{(t)}|\big), \tau_t = t/|\mathcal{A}_0|$.

f_θ is trained offline by behavioural cloning (supervised imitation learning) [22]. Rollouts follow Entropy with probability 0.75 and a random action otherwise; the supervised label is always Entropy's greedy choice, optimized by cross-entropy loss. We chose behavioural cloning over directly optimizing per-step nDCG improvement via reinforcement learning because the latter's reward is sparse and high-variance early in an elicitation trajectory, when most of the item ranking is still unresolved. Behavioural cloning instead supplies a dense supervised signal at every step by imitating Entropy's already-validated choice, at the cost of being bounded above by the expert it imitates. The mixed-rollout scheme (0.75 expert, 0.25 random) exposes the network to states an always-expert rollout would never visit, mitigating covariate shift between training and deployment [22]. The five-dimensional input in Eq. (4) was deliberately kept small. It excludes the raw item scores $s_i^{(t)}, s_j^{(t)}$, using only their sum and absolute difference, so the network cannot simply memorize item identities. This forces it to learn a transferable function of comparison state, a design choice consistent with the near-zero seen/unseen generalization gap reported in Section V-E. FlowClone tests whether Entropy's informativeness can be recovered from a compact, five-dimensional feature vector, and whether it generalizes to unseen users.

BALD is a Bayesian active-learning baseline [14]. Each item i has a Gaussian belief $\mathcal{N}(\mu_i^{(t)}, {\sigma_i^{(t)}}^2)$ under an online Bayesian Bradley-Terry model, updated after each comparison by a pairwise Gaussian-probit moment update adapted from [18], with fixed observation-noise scale $\beta = 0.5$. Expected information gain under this model is monotone in predictive variance, so BALD queries the pair of maximal combined variance,

$$\pi_{\text{bald}}(t) = \arg\max_{(i,j)} \; {\sigma_i^{(t)}}^2 + {\sigma_j^{(t)}}^2. \tag{5}$$

This is a standard tractable approximation that avoids Monte Carlo mutual-information estimation. We emphasize that we do not use the full TrueSkill system (teams, draws, or factor-graph scheduling); we only reuse its scalar Gaussian-probit update idea in a pairwise Bradley-Terry setting.

D. Scoring Models

Given acquired comparisons Q_t, item scores can be computed by **HodgeRank** (the online update of Eq. (2)), **Borda count** (a simple win-counting aggregator that ignores graph structure), **Win-Ratio** (the fraction of comparisons won, normalizing for count imbalance without exploiting Hodge structure), or **Bayesian Bradley-Terry** (the posterior mean $\mu_i^{(t)}$ from BALD's belief model). Separating acquisition policy from scoring model lets us isolate the contribution of each.

E. Evaluation Protocol

An early version of this protocol measured elicitation quality by Kendall correlation between the online ranking and a full-data batch ranking computed from the same user's comparisons. We identify this as circular: it measures how quickly a policy helps HodgeRank reconstruct its own eventual answer, not genuine held-out quality. We replace it as follows. Each user's comparisons are split 80/20 into P_{train} and P_{held}. Held-out relevance labels come from a full-data batch HodgeRank ranking over all of that user's comparisons, train and held-out combined, so ground truth is never a function of the online trajectory. This full-data ranking doubles as the protocol's implicit performance ceiling: nDCG@5 is computed against it directly, so a policy that recovered the full-data ranking exactly from the train pool alone would score nDCG@5 $= 1$, the same reference point every policy in Section V is measured against; no separate oracle policy is required to state this ceiling; it is already the metric's normalization. The item ranked r-th receives graded relevance $n_h - r$. After every elicitation step, the online score ranks the held-out items, and quality is measured by nDCG at cutoff $k = 5$ [23]. We summarize a trajectory by the trapezoidal area under the per-step nDCG@5 curve (AUC-nDCG) and by Comparisons-to-Threshold (CTT), the percentage of the train pool consumed before nDCG@5 first reaches $\tau = 0.80$.

IV. Experimental Setup

A. Datasets

We evaluate on three datasets (Table I) spanning a range of item-set size and comparison-graph density. Car1 users rank all 10 items ($\binom{10}{2} = 45$ implied comparisons); Sushi-10 users each rank a fixed 10-item subset of a shared 100-item universe; Sushi-X users rank a variable-length subset (5–10 items, mean 7.4). Under the five-fold leave-users-out protocol (Section IV-B), every user is evaluated exactly once, so the counts in Table I reflect the full population evaluated for every dataset.

TABLE I. Evaluation Datasets, Item-Set Sizes, User Counts, and Per-User Comparison-Graph Characteristics

Dataset	Items	Users	Characteristics
Car1 [24]	10	60	Dense, near-complete per-user graphs
Sushi-10 [25]	100	500	Sparse, uniform comparison count
Sushi-X [25]	100	500	Sparse, heterogeneous density

These three datasets span an order of magnitude in item-set size (10 to 100 items) and range from dense, near-complete per-user comparison graphs (Car1) to sparse, variable-length ones (Sushi-X), giving a deliberately heterogeneous test bed for the acquisition policies evaluated in Section V.

B. Implementation, Leave-Users-Out Protocol, and Statistics

Batch reference rankings use $\lambda = 10^{-5}$, solved via sparse LSQR. FlowClone's network is trained for 20 epochs at learning rate 10^{-3} on rollouts collected up to 60% of each user's train-pool budget. The Bayesian Bradley-Terry model uses prior variance $\sigma_0^2 = 1$ and $\beta = 0.5$, standard Gaussian-probit defaults in this pairwise setting, untuned for this study.

Each dataset's users are partitioned into five disjoint folds. For each fold, FlowClone's behavioural-cloning trajectories are collected exclusively from the other four-fifths of users, and every policy is evaluated only on that fold's disjoint held-out fifth, so every user is evaluated exactly once across folds. This gives full-population statistical power for every policy while preserving a genuine leave-users-out guarantee for FlowClone.

Each dataset $\times$ policy $\times$ scorer combination is evaluated over ten random seeds within every fold. Every user's per-seed AUC-nDCG values are averaged into a single paired observation for that user. We assess significance of each policy's advantage over Random (HodgeRank scorer) with a paired, two-sided Wilcoxon signed-rank test over these per-user values, matched by user across the full evaluated population (pooled across all five folds, since folds are disjoint in users). Ten seeds per fold comfortably clears the conventional $n = 6$ minimum for the test to have power. Pooling to the per-user level, rather than testing across seeds alone, gives the test a much larger effective sample size: one paired observation per user, not one per seed. The six policies $\times$ three datasets form a single family of 18 hypotheses, all testing the same claim (policy beats Random). We apply a Holm-Bonferroni step-down correction across all 18 raw p-values jointly, rather than per dataset, and report the Holm-adjusted p-value (p_{holm}) as our significance criterion in Table II. We also report the raw AUC-nDCG advantage Δ and a bootstrapped 95% confidence interval for Δ (10,000 resamples of the paired per-user differences), given inline in Section V wherever a result is discussed individually. We treat consistency of sign and magnitude across all three datasets as corroborating evidence alongside these statistics.

V. Results

A. Active Acquisition Roughly Halves the Comparison Budget

Table II reports per-user nDCG@5 at three budget fractions, AUC-nDCG, and CTT for the HodgeRank scorer, on the full population of each dataset. Entropy, Coverage, and FlowClone substantially and consistently outperform Random on all three datasets. Popularity-LB significantly underperforms Random on Car1 and significantly outperforms it on Sushi-10. Its point estimate on Sushi-X ($\Delta = +0.008$) is not statistically significant ($p_{\text{holm}} = 1$). On no dataset does it approach the geometry-informed policies.

Entropy, Coverage, and FlowClone reach nDCG@5 ≥ 0.80 using, on average, approximately **51% fewer comparisons** than Random (53% on Car1, 56% on Sushi-10, 43% on Sushi-X, via Entropy's CTT). All three improvements are consistent in sign and magnitude across every seed and fold on every dataset. Each remains significant after Holm-Bonferroni correction across the full 18-test family described in Section IV-B. Fig. 1 shows the full nDCG@5-versus-budget curves, Fig. 2 shows the same comparisons-to-threshold result as bars per policy and dataset, and Fig. 3 summarizes AUC-nDCG across all datasets and policies.

TABLE II. Main HodgeRank Results: Per-User nDCG@5 at 20%, 30%, and 50% Budget, Plus AUC-nDCG and CTT; Bold Marks Policies Significantly Better than Random after Holm Correction

Dataset	Policy	20%B	30%B	50%B	AUC	CTT (%)
Car1	Random	0.737	0.790	0.851	0.824	26.5
	Popularity-LB	0.627	0.701	0.799	0.769	35.3
	Entropy	**0.879**	**0.906**	**0.928**	**0.896**	**12.4**
	Coverage	**0.871**	**0.895**	**0.911**	**0.888**	**12.2**
	FlowClone	**0.866**	**0.900**	**0.925**	**0.891**	**13.2**
	HarmonicReduce	0.676	0.767	0.822	0.800	25.5
	BALD	0.771	0.821	0.886	0.850	24.2
Sushi-10	Random	0.770	0.812	0.875	0.841	27.2
	Popularity-LB	**0.743**	**0.870**	**0.941**	**0.886**	**21.1**
	Entropy	**0.884**	**0.950**	**0.956**	**0.923**	**12.0**
	Coverage	**0.884**	**0.947**	**0.969**	**0.926**	**12.1**
	FlowClone	**0.901**	**0.950**	**0.954**	**0.925**	**11.5**
	HarmonicReduce	0.791	0.847	0.859	0.849	9.1
	BALD	**0.820**	**0.788**	**0.896**	**0.878**	**19.1**
Sushi-X	Random	0.832	0.866	0.904	0.886	24.6
	Popularity-LB	0.776	0.836	0.908	0.894	23.9
	Entropy	**0.885**	**0.947**	**0.971**	**0.942**	**14.1**
	Coverage	**0.885**	**0.948**	**0.971**	**0.942**	**14.2**
	FlowClone	**0.896**	**0.953**	**0.971**	**0.944**	**13.7**
	HarmonicReduce	**0.904**	**0.935**	**0.952**	**0.937**	**12.3**
	BALD	**0.861**	**0.881**	**0.922**	**0.909**	**19.4**

Reporting comparison savings only as a percentage of each user's train pool (Table II, column CTT) obscures the absolute number of judgments this represents. Car1 users have a median 39.6 train-pool comparisons; Entropy's 12.4% CTT is therefore approximately 5 comparisons, against Random's 26.5% ($\approx$10–11 comparisons), an absolute saving of roughly 5–6 judgments per user. Sushi-10 users have a fixed 36 train-pool comparisons (80% of $\binom{10}{2} = 45$); Entropy's 12.0% CTT is approximately 4 comparisons against Random's 27.2% ($\approx$10), an absolute saving of roughly 6 judgments per user. Sushi-X users have a mean 18.9 train-pool comparisons (80% of a mean 23.7 implied pairs from a mean 7.4-item subset); Entropy's 14.1% CTT is approximately 3 comparisons against Random's 24.6% ($\approx$5), an absolute saving of roughly 2 judgments per user. The relative savings are therefore consistent (43–56%) while the absolute number of judgments saved per user is small at these item-set sizes (2–6 comparisons); the practical benefit compounds with the number of users elicited concurrently and would need re-measurement at the thousands-of-candidates scale, which this paper does not test (Section VI).

B. BALD: Active HodgeRank Policies Outperform a Genuine Bayesian Baseline

BALD tests whether HodgeRank-based acquisition offers a real advantage over a genuine Bayesian alternative, not only over Random and the Popularity-LB baseline. Table II shows BALD is a non-trivial baseline. It significantly outperforms

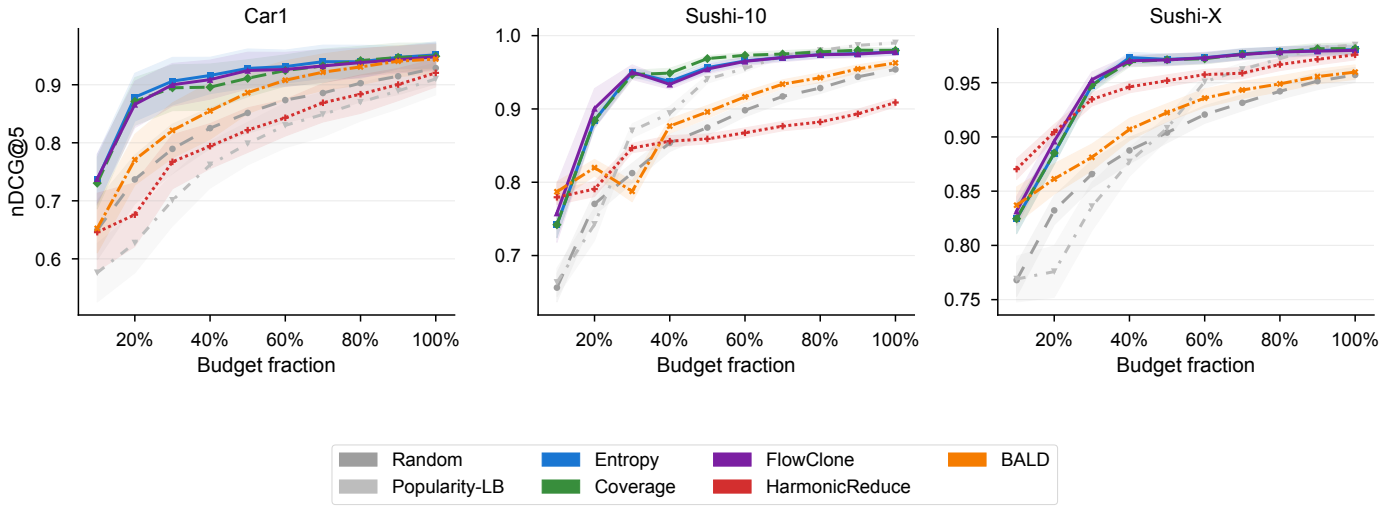


Fig. 1. nDCG@5 versus budget fraction for all acquisition policies under HodgeRank scoring (mean ± 1 standard deviation over seeds and folds).

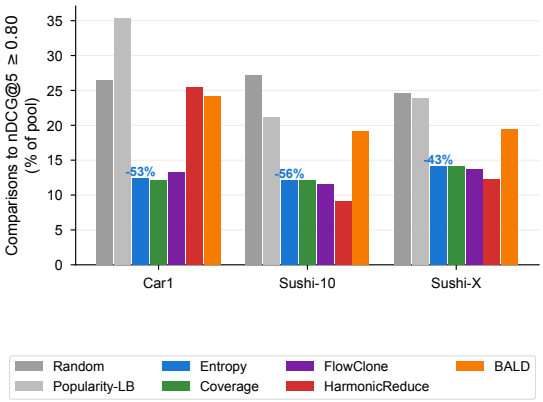


Fig. 2. Comparisons-to-threshold (CTT) to reach nDCG@5 ≥ 0.80, reported as a percentage of each user's training pool.

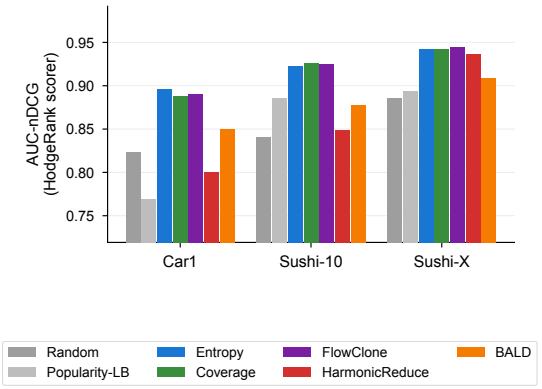


Fig. 3. AUC-nDCG summary across acquisition policies and datasets under HodgeRank scoring.

Random on Sushi-10 ($\Delta = +0.037$, $p_{\text{holm}} = 6.2 \times 10^{-17}$, $n = 500$) and Sushi-X ($\Delta = +0.023$, $p_{\text{holm}} = 9.9 \times 10^{-5}$, $n = 500$); both survive Holm-Bonferroni correction across the full 18-test family. On Car1, BALD's point estimate is smaller and does not reach significance ($\Delta = +0.026$, raw $p = 0.57$, $n = 60$), consistent with this dataset's much smaller evaluated population. Regardless of whether BALD's own advantage over Random is significant, Entropy, Coverage, and FlowClone consistently outperform BALD's point estimate on all three datasets, by roughly 0.03–0.06 AUC-nDCG and 5–12 points of CTT. We interpret this as reflecting what each criterion is sensitive to. BALD's posterior-variance criterion treats all belief uncertainty symmetrically, with no gradient/curl/harmonic distinction. Entropy and Coverage directly probe HodgeRank's structure instead.

C. HodgeRank is Necessary for Active Gains: The Borda Collapse

Table III reruns Entropy under all four scoring models. Entropy paired with Borda count is *worse* than Random paired with HodgeRank on every dataset, and worse than Random paired with Borda on every dataset as well.

The cause is structural. Entropy selects the comparison with the largest gradient residual, i.e. the one about which the current model is most surprised. Such a comparison is, almost by construction, one for which the two items have been nearly evenly split in prior wins and losses. Its Borda contribution, a simple win increment, is close to uninformative regardless of outcome. Entropy actively steers the query budget toward exactly the comparisons Borda is least equipped to exploit, producing outcomes worse than Random. We term this the Borda collapse. Active gains are not a generic property of any acquisition-plus-scoring pipeline; they depend on compatibility between the two. The failure mode is an outright reversal, not merely a smaller gain. Win-Ratio and Bayesian Bradley-Terry remain competitive with HodgeRank throughout. On the sparse Sushi datasets they are essentially tied with it or marginally ahead: HodgeRank trails Win-Ratio by 0.004 and Bayesian-BT by 0.010 on Sushi-10, and ties both to three decimal places on Sushi-X. We attribute this to the near-acyclic structure of sparse per-user graphs, where the harmonic component is

TABLE III. SCORER ABLATION UNDER ENTROPY ACQUISITION: AUC-NDCG FOR EACH SCORING MODEL, WITH Δ COMPUTED AGAINST RANDOM USING THE SAME SCORER. BOLD MARKS SIGNIFICANCE AFTER HOLM CORRECTION IN THE RELEVANT FAMILY ($\alpha = 0.05$)

Dataset	Scorer	Entropy	Random	Δ	p_{holm}
Car1	HodgeRank	0.896	0.824	**+0.072**	[a]
	Win-Ratio	0.860	0.827	**+0.033**	2.1×10^{-6}
	Bayesian-BT	0.889	0.843	**+0.046**	3.6×10^{-7}
	Borda	0.611	0.757	**−0.146**	1.2×10^{-10}
Sushi-10	HodgeRank	0.923	0.841	**+0.082**	[a]
	Win-Ratio	0.927	0.814	**+0.113**	1.2×10^{-82}
	Bayesian-BT	0.933	0.861	**+0.072**	2.3×10^{-80}
	Borda	0.628	0.764	**−0.136**	8.0×10^{-79}
Sushi-X	HodgeRank	0.942	0.886	**+0.056**	[a]
	Win-Ratio	0.942	0.861	**+0.081**	1.8×10^{-81}
	Bayesian-BT	0.942	0.897	**+0.045**	1.1×10^{-72}
	Borda	0.780	0.837	**−0.057**	5.0×10^{-31}

[a] see the 18-test family reported in Table II, in which every Entropy-vs-Random comparison is significant after Holm correction.

close to zero and Hodge decomposition adds little beyond win-counting or a Bayesian point estimate. On the denser Car1 graph, by contrast, HodgeRank holds a clear advantage over both alternatives (+0.036 over Win-Ratio, +0.007 over Bayesian-BT). This is consistent with HodgeRank's harmonic component only becoming substantial once the comparison graph carries enough cycles to require it. Fig. 4 visualizes this pattern across all three datasets.

Extending the same Wilcoxon-plus-Holm-Bonferroni procedure used for Table II to this ablation confirms that all nine Win-Ratio, Bayesian-BT, and Borda comparisons are significant, in both directions: the positive Win-Ratio and Bayesian-BT deltas and, notably, the negative Borda deltas that define the collapse described above. The Borda collapse is therefore tested against the same statistical standard as the paper's other headline comparisons; it is the largest-magnitude, most significant effect in this table on every dataset.

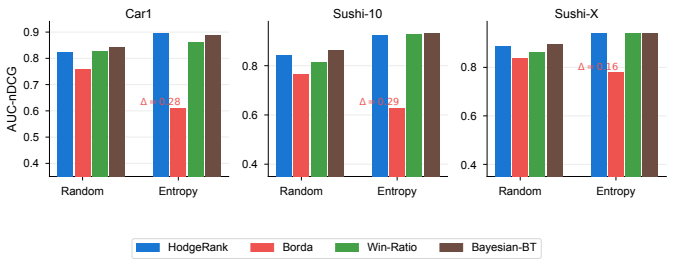


Fig. 4. Acquisition-scorer interaction for Random and Entropy; active gains persist for HodgeRank, Win-Ratio, and Bayesian-BT, while Borda exhibits the collapse.

D. HarmonicReduce: A Density-Dependent Hodge-Theoretic Finding

HarmonicReduce underperforms Random on Car1: a negative point estimate ($\Delta = -0.024$), significant under an uncorrected test (raw $p = 0.02$) but not significant once the full 18-test family is Holm-corrected ($p_{\text{holm}} = 0.08$). It shows dataset-density-dependent behavior on the sparse Sushi datasets, and this behavior differs between the two evaluation metrics. On Sushi-10, it reaches the nDCG@5 ≥ 0.80 threshold faster than every other policy tested (CTT 9.1% versus Random's 27.2%, a 3.0× reduction). But its AUC-nDCG advantage over Random is small and not statistically significant ($\Delta = +0.008$, raw $p = 0.67$, $p_{\text{holm}} = 1$, $n = 500$), and it trails Entropy's AUC-nDCG (0.849 versus 0.923) by a substantial margin. We read this as fast early convergence to a moderately good ranking that does not translate into a reliably higher AUC-nDCG over the full trajectory on this dataset. On Sushi-X, by contrast, HarmonicReduce's AUC-nDCG advantage over Random is both large and significant ($\Delta = +0.051$, $p_{\text{holm}} = 6.5 \times 10^{-28}$, $n = 500$). It closely matches Entropy (0.937 versus 0.942, a gap of only 0.005) while also reaching threshold faster (CTT 12.3% versus 14.1%), making it the clearly favorable policy on that dataset by both metrics. This is a direct, theoretically predicted consequence of Proposition 1: the degree-weighted proxy in Eq. (3) is discriminative only when item degree varies substantially, which occurs precisely when the graph is sparse. On Car1's near-complete graphs, item degrees are nearly uniform, and the proxy degenerates to a near-constant multiplier on the residual. HarmonicReduce therefore collapses toward a noisier Entropy without Entropy's benefit. Even this underperformance, while consistent in direction across every seed and fold we tested, is only raw-significant, not Holm-significant. This underscores how close to indifferent HarmonicReduce is to Random on this dense dataset once multiple comparisons are accounted for. On the sparse Sushi graphs, degree variance is substantial, and harmonic inconsistency is concentrated on specific high-degree edges. This makes the proxy genuinely discriminative early in elicitation and explains the CTT advantage on both Sushi datasets. Whether this early advantage survives to a significant full-trajectory AUC-nDCG gain differs between Sushi-10 and Sushi-X, an open question we do not resolve here. We report this transparently, including the two non-significant results on Car1 and Sushi-10. The explanation itself, not merely a single aggregate number, is the contribution: a Hodge-decomposition-derived heuristic's practical value, and its limits, can be predicted in advance from approximate comparison-graph completeness. Its statistical reliability should be assessed metric by metric, under appropriate multiple-comparisons correction, rather than assumed to transfer from one metric or dataset to another.

Fig. 5 shows this contrast directly.

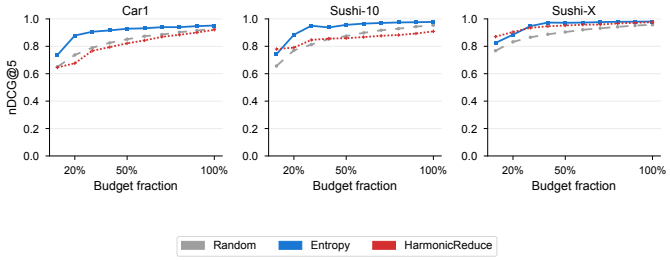


Fig. 5. HarmonicReduce compared with Random and Entropy across datasets, showing its density-dependent behavior in both early and full-budget regimes.

E. FlowClone: An Offline-Trainable Approximation of Entropy

Despite its compact five-dimensional feature representation, FlowClone's AUC-nDCG is statistically indistinguishable from Entropy's on all three datasets under our leakage-free, five-fold leave-users-out protocol (-0.005 on Car1, $+0.002$ on Sushi-10, $+0.002$ on Sushi-X; Table II). FlowClone is the only policy that consumes training data. We therefore additionally compared, pooled across all five folds, its AUC-nDCG on users it was trained on versus the disjoint users it was evaluated on in the same fold. The seen-versus-unseen gap , shown in Fig. 7, shrinks to near zero and changes sign across datasets (Car1: $\Delta = +0.0035$; Sushi-10: $\Delta = -0.0005$; Sushi-X: $\Delta = -0.0017$, seen minus unseen). Every gap is smaller than the corresponding standard deviation of per-user AUC-nDCG. This is read as population-level evidence that FlowClone does not overfit to its training users: what it learns transfers, within noise, to entirely unseen users in every fold and dataset tested. Fig. 6 shows the corresponding training-loss curves converge smoothly on all three datasets, with no sign of instability that would cast doubt on the trained network used for this comparison.

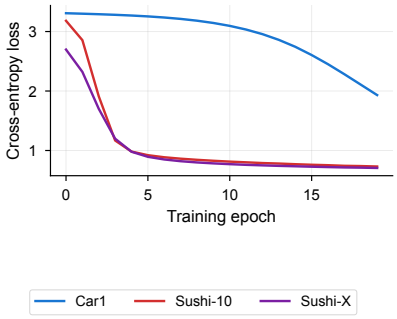


Fig. 6. FlowClone cross-entropy training loss by epoch (fold 1) for Car1, Sushi-10, and Sushi-X.

F. Robustness to the Convergence-Threshold Definition

CTT and the headline efficiency figure both depend on the choice of convergence threshold $\tau = 0.80$. We therefore swept τ from 0.65 to 0.90 and recomputed CTT for Random, Entropy, and Coverage on all datasets, pooled across all five folds (Fig. 8). The Random-to-Entropy efficiency ratio remains within $1.3\times$–$2.8\times$ across the entire tested range and all datasets. This confirms the central efficiency finding is not an artefact of the particular threshold chosen for the headline result.

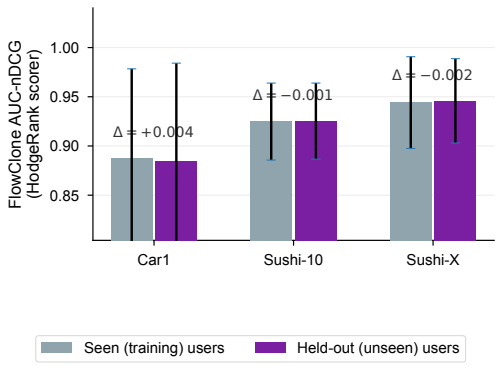


Fig. 7. FlowClone AUC-nDCG on seen (training) versus unseen (held-out) users, pooled across five folds, with per-dataset generalization gaps.

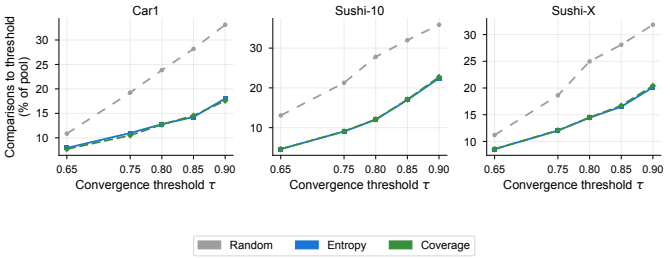


Fig. 8. CTT sensitivity to the convergence threshold τ for Random, Entropy, and Coverage across all datasets.

G. The Online Update Tracks the Batch Estimator

Section III-B frames the online update (Eq. (2)) as a stochastic-approximation scheme for the batch objective (Eq. (1)). This is verified empirically as follows: for a random (non-adaptive) query order, holding out no items, the online running score after each step is compared by Kendall τ against the batch HodgeRank solution computed over the same user's full comparison set. Random order isolates the online/batch relationship itself from any acquisition-policy effect. Averaged over 40 randomly sampled users per dataset and 5 seeds each, τ rises monotonically with the fraction of comparisons observed and reaches 0.90 (Car1), 0.95 (Sushi-10), and 0.95 (Sushi-X) once all comparisons are incorporated, up from 0.38–0.43 at 10% of comparisons on all three datasets. The online estimate does not reach exact agreement with the batch solution even at 100% of comparisons, which is expected: the online update has no explicit ℓ_2 regularization term and follows a single online pass rather than jointly solving the weighted least-squares system, so the two estimators are not guaranteed to coincide exactly at any finite step, only to converge together as the stochastic-approximation conditions on η_t are satisfied. This confirms the online update is a genuine, increasingly faithful approximation of the batch estimator.

H. Computational Overhead of Acquisition Policies

Reporting only relative comparison savings leaves open whether an acquisition policy is itself cheap enough to deploy at the per-decision latency a live system requires. Table IV reports the wall-clock cost of the selection step alone (excluding the shared, policy-independent cost of maintaining OnlineHodgeRank and OnlineBayesianBT, which every policy pays identically), averaged over 200 timed calls per candidate-pool size after 20 untimed warm-up calls, at pool sizes spanning this paper's datasets (10 items $\Rightarrow$ up to 45 candidate pairs at Car1's density) up to a synthetic 2,000-candidate pool representative of a substantially larger catalogue.

TABLE IV. MICROSECONDS PER ACQUISITION DECISION (SELECTION STEP ONLY) BY CANDIDATE-POOL SIZE

Policy	@10	@45	@100	@500	@2000
Entropy	2.8	12.3	25.4	127.4	493.1
Coverage	9.6	32.4	63.3	292.7	1720.5
HarmonicReduce	9.7	35.7	67.5	289.1	1168.1
BALD	2.3	9.5	20.0	98.2	392.4
FlowClone	35.0	100.7	201.0	908.8	4885.3
FlowClone / Entropy	12.4×	8.2×	7.9×	7.1×	9.9×

FlowClone is measured to be 7–12× slower per decision than Entropy at every pool size tested (single-run wall-clock measurement). The exact ratio fluctuates modestly across repeated runs due to timing noise at microsecond scale, but the qualitative result is stable across runs. The measured cost does not come from the network itself, a 5-64-64-1 MLP forward pass is a handful of small matrix multiplications, but from constructing its five-dimensional feature vector (Eq. (4)) for every candidate pair, which includes a common-neighbour set intersection per candidate, the same operation that makes Coverage and HarmonicReduce several times slower than Entropy in the same table. FlowClone pays this cost per candidate in addition to, not instead of, Entropy's own residual computation, since its feature vector includes the residual as one input. This is an implementation-dependent cost of per-candidate feature construction, not an inherent property of the distilled network, and a vectorized batch implementation of the feature computation may substantially close this gap. FlowClone is not currently a faster substitute for Entropy, and its practical justification rests on distillation fidelity and cross-user generalization (Section V-E), not on deployment latency.

VI. DISCUSSION

The roughly 51% comparison-budget reduction is consistent across three structurally different datasets. This suggests a concrete deployment guideline: systems collecting pairwise preferences for HodgeRank-based recommendation should adopt gradient-residual (Entropy) or common-neighbour (Coverage) acquisition by default. Density-dependent policies such as HarmonicReduce should be reserved for settings where the comparison graph is known in advance to be far from complete. The BALD baseline confirms this advantage is not merely because HodgeRank-based policies beat weak or naive baselines: a properly implemented Bayesian active learner is also outperformed. Coverage additionally requires no trained model, making it a natural default where implementation simplicity matters as much as query efficiency. FlowClone is attractive as a distilled approximation of Entropy that preserves near-identical ranking quality under leave-users-out evaluation, though Section V-H shows this distillation is not currently a deployment-latency win: at every pool size tested, FlowClone is 7–12× slower per decision than the Entropy policy it approximates, so its practical justification rests on model compactness and cross-user transfer rather than on speed. These properties map onto distinct deployment scenarios. Cold-start onboarding, where a new user's item preferences must be elicited under a strict comparison budget before a first recommendation can be produced, benefits directly from the roughly halved comparisons-to-threshold shown in Table II. Real-time or high-throughput settings, where many users are elicited concurrently, benefit from Coverage's simple hand-crafted rule; FlowClone remains attractive if a compact, transferable model of Entropy's behaviour is independently valuable, for instance for on-device deployment where Entropy's own residual computation is unavailable.

The Borda collapse documents the case of an active acquisition policy that is actively harmful, not merely less beneficial, when paired with an incompatible scoring model. This has a direct methodological implication: reporting an acquisition policy's gain over random selection is insufficient without specifying, and justifying, the compatibility of the downstream aggregation model. The HarmonicReduce density-dependence finding similarly shows that a heuristic's practical value can be predicted, not merely observed post hoc, from graph-theoretic properties that Hodge decomposition makes explicit. The concurrent RLHF-motivated active preference literature operates on scalar Bradley–Terry-style reward models. Our direct implementation of a representative Bayesian Bradley-Terry active learner (BALD), which HodgeRank-based acquisition outperforms, together with our scorer ablation, suggests that acquisition-scorer compatibility deserves explicit examination rather than being assumed.

The BALD comparison points to a plausible explanation for why geometry-informed acquisition holds an advantage over a well-specified Bayesian alternative. BALD's posterior-variance criterion treats belief uncertainty as a single scalar per item. It therefore cannot distinguish a comparison that would resolve a locally inconsistent triangle (curl) from one that would resolve a globally unresolvable disagreement (harmonic) from one that would simply sharpen an already largely settled ranking (gradient). All three reduce posterior variance and are treated interchangeably. Entropy and Coverage, by contrast, operate directly on the residual of the gradient fit. This residual is itself informative about where the current consensus ranking is least supported by observed comparisons. This is consistent with the pattern in Table II: BALD's advantage over Random is real but consistently smaller than Entropy's or Coverage's, across every dataset where BALD itself is unambiguously informative.

We highlight several limitations. This paper targets the item-set scale typical of recommender systems and group decision-making (10–100 items), not the thousands-of-candidates scale of some large language model reward-modeling pipelines. Within this scale, our mechanism and theoretically grounded predictions (density dependence, scorer compatibility) are not directly validated beyond the 100-item Sushi setting. Evaluation follows a leakage-free offline-

replay protocol on previously collected comparison datasets (Section III), consistent with prior active-ranking work [15]. The HarmonicReduce density-dependence claim rests on a two-point (dense-versus-sparse) contrast, not a continuous density sweep. This is an opportunity for dedicated follow-up. FlowClone's feature representation is intentionally compact, to test whether a handful of simple statistics suffice. A larger feature set might close its small residual gap to Entropy on Sushi-X, but would weaken the distillation claim, so we did not pursue it. Several hyperparameters, most notably γ_0 (the online update's warm-up constant), λ (HodgeRank's regularization strength), and the Bayesian Bradley-Terry model's prior and noise parameters, are fixed per dataset rather than swept systematically; a targeted stability check would confirm the reported comparison-budget reductions are not artifacts of these particular settings, and is left to future work rather than attempted here for every hyperparameter family.

Several extensions follow naturally. A continuous comparison-graph-density sweep rather than the two-point dense/sparse contrast used here would more precisely characterize where the HarmonicReduce transition occurs, and test whether it follows a sharp threshold or a gradual trend, guided directly by Proposition 1. Extending to batch, rather than single-comparison, acquisition would require selecting a diverse set of informative pairs per round instead of the single greedy pair each policy currently selects; this matters for item catalogues large enough that per-comparison model updates are impractical. A systematic study of acquisition-scorer compatibility across the broader family of pairwise aggregation models used in recommender systems and group decision-making would extend the four-scorer ablation in Section V-C to rank-centrality and matrix-factorization-based aggregators. A further direction, suggested by the BALD comparison, is a hybrid acquisition rule that combines HodgeRank's gradient residual with an explicit posterior-variance term. Such a rule could capture both the structural informativeness Entropy and Coverage exploit and the calibrated uncertainty estimate BALD provides.

VII. Conclusion

This paper asked whether the Hodge-theoretic structure used elsewhere for aggregation and quality assessment is also informative for deciding which pairwise comparisons to actively request. We use a leakage-free, five-fold leave-users-out evaluation protocol and seven acquisition policies, evaluated across three real-world datasets and four scoring models. Geometry-informed active acquisition (Entropy, Coverage, and its lightweight behavioural-cloning approximation FlowClone) reduces the comparison budget required to reach a fixed recommendation-quality threshold by approximately 51% relative to random sampling. It consistently outperforms a properly implemented Bayesian Bradley-Terry active learner (BALD) on all three datasets; BALD itself is a statistically confirmed improvement over random sampling on two of the three. A novel harmonic-energy-targeted policy, HarmonicReduce, exhibits a dataset-density dependence that we both observe empirically and prove exactly (Proposition 1) from the circuit rank of the comparison graph. FlowClone is validated under five-fold cross-validation, with a seen-versus-unseen generalization gap that is near zero and changes sign across datasets, indicating no detectable overfitting. Taken together, these findings indicate that combinatorial Hodge theory provides a coherent geometric language across the pairwise-preference pipeline, from deciding what to ask to aggregating what was said.

Conflict of Interest Statement

The authors declare that there is no conflict of interest.

Funding

This research received no external funding.

Declaration on Generative AI

Generative AI was used solely for language refinement and grammatical improvement. The authors take full responsibility for the final content of the manuscript.

Acknowledgments

The authors thank the maintainers of the Car Preference [24] and Sushi [25] datasets used throughout this work.

References

[1] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4299–4307, 2017.

[2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[3] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[4] J. Rothe, "Borda count in collective decision making: A summary of recent results," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 9830–9836, 2019.

[5] W. Muldrew, P. Hayes, M. Zhang, and D. Barber, *Active preference learning for large language models*, ser. ICML'24. JMLR.org, 2024.

[6] N. Das, S. Chakraborty, A. Pacchiano, and S. R. Chowdhury, *Active Preference Optimization for Sample Efficient RLHF*, R. P. Ribeiro, B. Pfahringer, N. Japkowicz, P. Larrañaga, A. M. Jorge, C. Soares, P. H. Abreu, and J. Gama, Eds. Cham: Springer Nature Switzerland, 2026.

[7] K. Ji, J. He, and Q. Gu, "Reinforcement learning from human feedback with active queries," *arXiv preprint arXiv:2402.09401*, 2024.

[8] Y. Shen, H. Sun, and J.-F. Ton, "Reviving the classics: Active reward modeling in large language model alignment," *arXiv preprint arXiv:2502.04354*, 2025.

[9] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, "Statistical ranking and combinatorial Hodge theory," *Mathematical Programming*, vol. 127, no. 1, pp. 203–244, 2011.

[10] Q. Xu, Y. Yao, X. Jiang, and Q. Huang, "HodgeRank on random graphs for subjective video quality assessment," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 844–857, 2012.

[11] E. Bejaoui, M. O.-E. Aoueileyine, R. Bouallegue, and A. Yazidi, "Hodge-based methods for contributor quality assessment in crowd-sourced preferences," in *Advanced Information Networking and Applications (AINA 2026)*, L. Barolli, W. K. G. Seah, and I. Woungang, Eds. Cham: Springer Nature Switzerland, 2026, pp. 260–272.

[12] B. Settles, "Active learning literature survey," *Computer Sciences Technical Report 1648, University of Wisconsin–Madison*, 2009.

[13] W. Chu and Z. Ghahramani, "Preference learning with Gaussian processes," *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 137–144, 2005.

[14] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[15] K. G. Jamieson and R. D. Nowak, "Active ranking using pairwise comparisons," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2240–2248, 2011.

[16] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3–4, pp. 324–345, 1952.

[17] E. Bejaoui, M. O.-E. Aoueileyine, R. Bouallegue, and A. Yazidi, "Hodgerank for group decision-making: Evaluating recommendation stability with incomplete preference data," in *2025 2nd International Conference on Innovative and Intelligent Information Technologies (IC3IT)*. IEEE, 2026, pp. 1–6.

[18] R. Herbrich, T. Minka, and T. Graepel, "Trueskill™: a bayesian skill rating system," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, p. 569–576.

[19] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, 1982.

[20] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[21] R. Diestel, *Graph Theory*, 5th ed. Berlin, Heidelberg: Springer, 2017.

[22] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 15, 2011, pp. 627–635.

[23] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[24] E. Abbasnejad, S. Sanner, E. V. Bonilla, and P. Poupart, "Learning community-based preferences via Dirichlet process mixtures of Gaussian processes," *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1213–1219, 2013.

[25] T. Kamishima, "Nantonac collaborative filtering: Recommendation based on order responses," *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 583–588, 2003.

EfficientNetB0-Based DeepFake Image Detection: A Transfer-Learning Framework for Robust and Deployable Media Verification

Marwa Hamza*, Mohammad Abdallah
Computer Science Department-Faculty of Science and IT, Al-Zaytoonah University of Jordan, Amman, Jordan[1,2]

***Abstract*—As the realism of DeepFakes has increased significantly, there is an emerging necessity to develop efficient, reliable, and robust automatic detection frameworks that are feasible to apply in the wild. This study proposes an image-based DeepFake detection system using EfficientNetB0 and transfer learning. Facial images were acquired from the FaceForensics++ and Celeb-DF datasets and preprocessed as 224 by 224 RGB images. The data were augmented to reflect realistic conditions through rotation, zoom, horizontal flipping, illumination changes, and JPEG compression. EfficientNetB0 was used as a feature extractor, followed by global average pooling, a dense layer with 128 neurons, dropout, and a two-class softmax layer. The model achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score. Robustness testing produced 89.5% accuracy on images compressed at JPEG quality 70, while cross-dataset evaluation from FaceForensics++ to Celeb-DF produced 84.2% accuracy. A real-time verification system was implemented by integrating the model with a Flask RESTful API and web and mobile clients. The findings show that transfer learning provides a practical balance among accuracy, robustness, computational efficiency, and deployability. However, the reduction in cross-dataset performance indicates limited generalizability to previously unseen manipulation techniques.**

***Keywords*—DeepFake detection; EfficientNetB0; transfer learning; convolutional neural networks; FaceForensics++; Celeb-DF; digital forensics**

I. Introduction

The evolution of artificial content creation has brought about a radical and increasingly visible transformation. Early face-swapping techniques were relatively simple manipulation tools; however, recent advances in deep generative models have enabled the production of highly realistic synthetic images with increasingly convincing visual details [1], [2]. This rapid development has raised serious concerns in multimedia forensics, particularly regarding the reliability of digital evidence and the misuse of manipulated media for misinformation, identity impersonation, and fraudulent activities [3]. Moreover, the growing realism of synthetic content may undermine public trust in digital media and online information [4].

DeepFake detection has consequently become a major challenge in multimedia forensics. One of the key difficulties is the ability of detection models to generalize to newly emerging generation techniques whose visual artifacts may differ from those observed during training [2], [5]. Furthermore, common image transformations and social-media processing, such as compression and resizing, may suppress or alter forensic traces used by detection algorithms [5]. Therefore, a reliable DeepFake detection framework should combine effective visual representation learning with computational efficiency and robustness against common image-quality degradations [1], [5].

Deep neural networks have been among the most successful methods for visual forgery detection, mainly because convolutional neural networks (CNNs) can extract subtle texture and boundary details that are difficult to encode manually. FaceForensics++ provides a large benchmark dataset containing multiple manipulation methods and compression levels [1]. Celeb-DF, by contrast, provides higher-quality DeepFakes that more closely resemble content encountered on social-media platforms [2].

ResNet [4] and Xception [3] are advanced and effective visual representation architectures. At the same time, practical systems must account for inference cost. EfficientNet addresses this need by scaling network depth, width, and input resolution jointly rather than independently [5].

This paper extends the DeepFake detector with an experimental and deployment-oriented evaluation. Its principal contributions are as follows:

- A transfer-learning application using EfficientNetB0 to detect DeepFakes in static facial images, together with a data-augmentation methodology that models distortions common to image-sharing channels.
- An evaluation of accuracy, precision, recall, F1-score, confusion-matrix performance, robustness to compression, and cross-dataset generalization.
- A RESTful DeepFake detection service that exposes the trained model through web and mobile clients and demonstrates an end-to-end workflow from preprocessing to live inference.

The initial detector achieved more than 92% validation accuracy on the evaluated face images and demonstrated a complete workflow from data preprocessing to live inference through the DeepFake Detector website.

II. Related Work

A. DeepFake Generation and Detection

Generative adversarial networks (GANs) consist of a generator and discriminator trained adversarially to produce increasingly realistic images [6]. Subsequent generative methods have

*Corresponding author.

improved identity transfer, reenactment, and facial synthesis, making genuine and fabricated content harder to distinguish. Early DeepFake detection methods often relied on visible artifacts or physical irregularities, but these cues became less dependable as generation quality improved.

Current detection systems largely depend on CNNs and related architectures for representation learning [7]–[10]. Rössler et al. introduced the FaceForensics++ dataset, which contains samples generated using DeepFakes, Face2Face, FaceSwap, and NeuralTextures at multiple compression levels [1]. Their work highlighted the importance of data-driven detection under compression.

Li et al. later released Celeb-DF, a database of higher-quality synthetic videos with fewer obvious artifacts and more challenging evaluation conditions [2]. The two datasets are complementary: FaceForensics++ supports detailed analysis across manipulation methods, while Celeb-DF challenges detectors to generalize to more visually convincing forgeries.

B. CNN Backbones and Efficient Architectures

CNN backbones such as ResNet and Xception are widely used in image classification and forensic detection [3], [4]. Xception reduces computational cost while preserving representational capacity through depthwise separable convolutions [3]. EfficientNet extends the efficiency objective by scaling depth, width, and resolution jointly through a compound coefficient. EfficientNetB0 is the smallest baseline in the family and offers a useful balance between accuracy and efficiency for API-based or mobile-connected deployment [5].

C. Transfer Learning and Robustness

Transfer learning is particularly valuable when a target dataset is smaller or more specialized than the large corpora used to pretrain modern CNNs. A pretrained model supplies generic visual filters at multiple levels, while task-specific modules are optimized for the real-versus-fake decision. Fine-tuning selected upper layers adapts the representation to the target problem while reducing the training difficulty and cost associated with learning a model from scratch.

Robustness is a major concern in DeepFake detection because resizing, JPEG compression, illumination differences, and image recapture can modify forensic cues. Data augmentation exposes the detector to such transformations during training and reduces sensitivity to a narrow set of acquisition conditions [1], [7], [8].

Recent image-classification studies also support the use of deep architectures, transfer learning, and augmentation in visually complex tasks. Vision Transformer benchmarking has shown that augmentation can strengthen image classification under varying visual conditions [14]. End-to-end segmentation and classification architectures have demonstrated effective feature learning in challenging medical images [15], while automated disease detection has shown the practical value of deep-learning pipelines for robust image-based recognition [16].

The rapid growth of AI-generated content has extended the digital-authenticity problem beyond manipulated faces. Deep-learning models using acoustic features such as mel-frequency cepstral coefficients have been applied to synthetic-audio detection [11], transformer-based approaches have been used for Arabic fake-news detection [12], and multimodal hybrid models have been investigated for automated fake-news verification [13]. These studies show that content verification is increasingly a multimodal challenge spanning visual, audio, and textual information.

D. Gap and Study Positioning

A persistent challenge in DeepFake research is the discrepancy between strong performance within the training dataset and degraded accuracy on previously unseen manipulation methods. In addition, many studies emphasize model metrics while giving limited attention to deployability. This study evaluates the detector through baseline classification, JPEG compression, and cross-dataset transfer while also implementing an end-to-end application. The combined setting examines both controlled performance and practical operation.

III. Materials and Methods

A. Experimental Dataset

The experiment uses the FaceForensics++ and Celeb-DF datasets. FaceForensics++ is a facial-manipulation benchmark containing multiple manipulation methods and compression settings [1]. Celeb-DF provides higher-quality, visually convincing DeepFakes created from celebrity footage [2]. Frames extracted from these sources form a two-class image-classification problem with Real and Fake labels.

FaceForensics++ provides the primary development data, while Celeb-DF is used as an external cross-dataset benchmark to assess generalization to higher-quality manipulations. To minimize correlation among samples, experimental partitioning should be performed at the source-video level before frame extraction, preventing frames from the same source video from appearing in both training and evaluation partitions. The final experimental record should report source-video counts, extracted-frame counts, class distributions, manipulation categories, and partition sizes.

B. Image Preprocessing and Augmentation

Each input image is converted to RGB, resized to 224×224 pixels, converted to a floating-point array, normalized, and expanded to batch format before model inference. During training, augmentation is applied to improve generalization and simulate transformations common to image-sharing environments. The augmentation operations and parameters are summarized in Table I.

C. EfficientNetB0 Classification Model

The detector is implemented using the Keras Functional API. EfficientNetB0 is used as the pretrained feature extractor, followed by a classification head comprising global average pooling, a fully connected layer with 128 units and ReLU activation, dropout with a rate of 0.40, and a final dense layer with two softmax outputs. The custom head contains 164,226 trainable parameters.

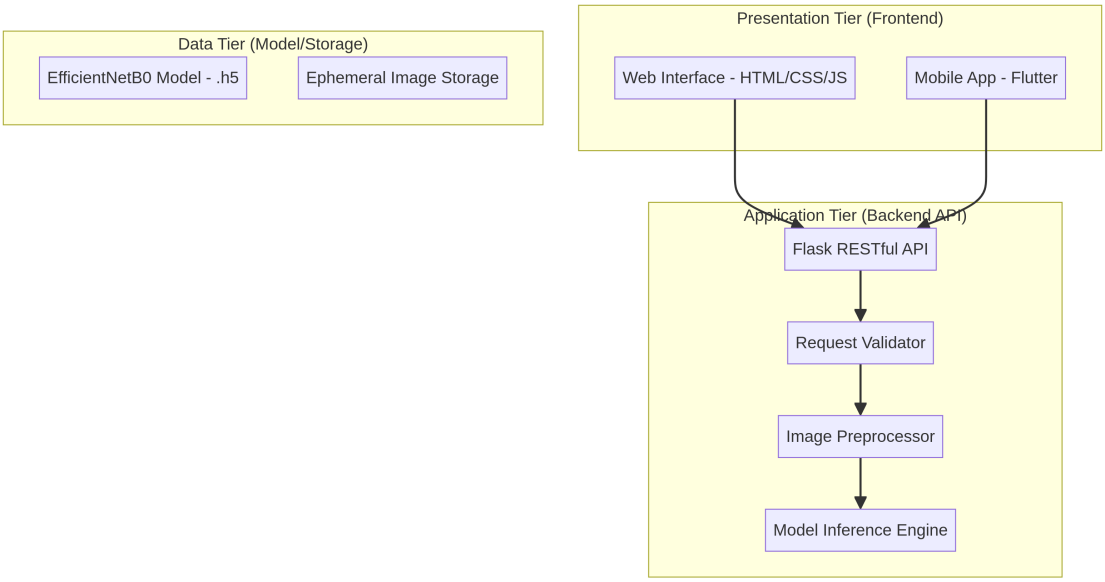


Fig. 1. Three-tier architecture of the implemented DeepFake detection system.

TABLE I. DATA AUGMENTATION STRATEGY USED IN THE EXPERIMENT

Augmentation	Purpose	Parameter
Random rotation	Simulate camera-angle variation	$\pm 10^{\circ}$
Random zoom	Model subject-distance variation	0.8–1.2
Horizontal flip	Account for mirrored images	50% probability
Brightness adjustment	Model illumination changes	$\pm 20\%$
JPEG compression	Model transmission and social-media loss	Quality 70–100

For input x and model parameters θ, the softmax layer estimates posterior scores for the Real and Fake classes. The predicted class is selected according to:

$$\hat{y} = \underset{c \in \{\text{Real},\text{Fake}\}}{\arg\max} \; p(y = c \mid x; \theta). \tag{1}$$

D. Training Strategy

Training proceeds in two stages. First, the EfficientNetB0 backbone is frozen and only the custom classification head is trained. This allows the classifier to learn a task-specific decision boundary without changing the pretrained visual representation. Second, selected upper layers are unfrozen and optimized using a lower learning rate. This two-stage procedure suppresses training fluctuations and adapts the model while limiting disruption to pretrained features.

E. End-to-End System Architecture

The experimental setup uses the three-tier application architecture shown in Fig. 1. The presentation tier contains a responsive web interface and a Flutter mobile application. The application tier uses a Flask RESTful API to validate requests, coordinate preprocessing, perform model inference, and format results. The data/model tier hosts the trained EfficientNetB0 model and temporary image storage. This separation allows the model to be maintained independently from the user-facing applications and supports practical deployment.

F. Evaluation Protocol and Metrics

The detector is evaluated using accuracy, precision, recall, and F1-score. In the binary setting, Fake is treated as the positive class. Accuracy measures the proportion of correct decisions, precision measures the fraction of predicted fakes that are manipulated, recall measures the fraction of manipulated images detected, and the F1-score is the harmonic mean of precision and recall. The latter three metrics are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{3}$$

$$F_1 = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{4}$$

Two additional experiments examine robustness. First, JPEG compression is applied at multiple quality levels, including a quality factor of 70. Compression removes image detail as quality decreases and can suppress forensic traces. Second, cross-dataset generalization is evaluated by training on FaceForensics++ and testing on Celeb-DF. These experiments expose common sources of performance degradation in practical DeepFake detection systems.

G. Software Deployment and Functional Testing

The inference service is developed in Python using TensorFlow, Keras, and Flask. The API accepts JPEG and PNG images, preprocesses each upload according to the model's requirements, performs inference, and returns the predicted label with a confidence score.

The functional test suite includes real-image uploads, GAN-generated images, low-resolution images, and concurrent requests. Ten uploads were handled simultaneously within five seconds, demonstrating the operational performance of the software implementation.

IV. Experimental Results

A. Overall Classification Performance

The final EfficientNetB0 model achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score, as summarized in Table II. These values are calculated from the 2,000-sample confusion matrix in Table III, ensuring consistency between aggregate metrics and class-level outcomes.

TABLE II. Main Validation Performance of the EfficientNetB0 Detector

Metric	Result
Validation accuracy	92.8%
Precision	92.5%
Recall	93.1%
F1-score	92.8%

B. Confusion-Matrix Analysis

Table III presents the validation confusion matrix, with 925 authentic samples correctly classified as Real, 75 authentic samples classified as Fake, 931 manipulated samples correctly classified as Fake, and 69 manipulated samples classified as Real. Based on these 2,000 samples, accuracy is 92.8%, precision is 92.5%, recall is 93.1%, and F1-score is 92.8%.

TABLE III. Confusion Matrix for the Validation Experiment

	Predicted Real	Predicted Fake
Actual Real	925	75
Actual Fake	69	931

C. Robustness to JPEG Compression

Compression is highly relevant to practical DeepFake detection because social-media and messaging platforms frequently recompress uploaded images. As shown in Table IV, accuracy decreased from 92.8% under the baseline condition to 89.5% at JPEG quality factor 70, an absolute reduction of 3.3 percentage points. The detector therefore retains substantial classification performance under moderate JPEG compression, although compression can alter forensic cues used by the classifier.

D. Cross-Dataset Generalization

A more challenging evaluation trained the detector on FaceForensics++ and tested it on Celeb-DF. Cross-dataset accuracy reached 84.2%, which is 8.6 percentage points below the 92.8% within-validation result. The decrease demonstrates the effect of dataset shift and highlights the difficulty of transferring a detector to visually different manipulation pipelines.

TABLE IV. Robustness Under JPEG Compression

Condition	Accuracy	Absolute Change
Uncompressed/baseline	92.8%	–
JPEG quality = 70	89.5%	−3.3 pp

TABLE V. Functional and Performance Test Cases

Test	Expected Behavior	Reported Outcome
Real image upload	Classify as Real	Pass
GAN-generated image	Classify as Fake	Pass
Low-resolution image	Return Real/Fake with reduced certainty	Pass; lower confidence
10 concurrent uploads	Process all requests in less than 5 s	Pass

Celeb-DF was designed to contain higher-quality, more realistic DeepFakes with fewer obvious artifacts [2], making it an appropriate benchmark for assessing generalization beyond the development dataset.

E. Application-Level Testing

The model was evaluated at the application level using the functional scenarios summarized in Table V. Tests covered genuine-image uploads, GAN-generated images, low-resolution inputs, and concurrent requests. Low-resolution images were processed with reduced confidence, while ten concurrent uploads were completed within the five-second target. These tests demonstrate the integration of the trained detector within a complete software pipeline; they do not replace statistical model assessment.

V. Discussion

A. Effectiveness of Transfer Learning

The findings support EfficientNetB0 as a feasible lightweight backbone for binary DeepFake image recognition. The validation accuracy of 92.8% is consistent with the use of pretrained CNN models as feature extractors that can be fine-tuned for image-forgery detection. From a deployment perspective, EfficientNetB0 is attractive because its compound-scaling design controls both accuracy and computational cost [5]. The custom head is also small, comprising global average pooling, a 128-unit dense layer, dropout, and the classifier, which limits the number of task-specific trainable parameters.

B. Role of Data Augmentation

The JPEG-compression experiment is consistent with the intended role of augmentation in exposing the model to common image transformations during training. At quality factor 70, accuracy decreased by 3.3 percentage points relative to the baseline. A dedicated ablation experiment is needed to isolate the contribution of JPEG augmentation from rotation, zoom, horizontal flipping, brightness variation, dropout, and fine-tuning.

C. Generalization Remains the Main Challenge

The cross-dataset accuracy of 84.2% is lower than the within-validation result, but it is an important finding rather than merely a limitation. DeepFake detection models can learn dataset-specific acquisition patterns, compression signatures, or generator artifacts. When the synthesis pipeline changes, these signals may disappear. Celeb-DF was designed to contain more naturalistic DeepFakes with fewer visible artifacts [2], which helps explain the observed decrease. Future studies should report and compare both within-dataset and cross-dataset results across newer generation families.

D. Comparison with Related Approaches

This study selected EfficientNetB0 because heavier backbones can achieve strong benchmark results at a higher computational cost. Xception remains a widely used architecture for DeepFake detection and uses depthwise separable convolutions [3], while ResNet introduced residual connections that support very deep visual models [4].

The proposed framework investigates whether EfficientNetB0 can provide competitive DeepFake classification performance while remaining suitable for API-based and mobile/web deployment. A controlled benchmark using identical training and test partitions for EfficientNetB0, Xception, and ResNet would provide a stronger basis for quantifying the accuracy-efficiency trade-off.

E. Experimental Validation Requirements

Further evaluation should include ablation studies of EfficientNetB0, transfer learning, fine-tuning, dropout, and individual augmentation operations; robustness tests for rotation, zoom, brightness variation, and JPEG compression; repeated runs summarized using means, standard deviations, and confidence intervals; and deployment measurements including total parameters, floating-point operations, model size, memory use, preprocessing overhead, inference latency, throughput, and hardware configuration. Controlled comparison with alternative backbones on identical data partitions would further support the accuracy-efficiency analysis.

F. Deployment Considerations

Deployment introduces constraints that are often omitted from benchmark-only studies. The detection service must validate files, transform inputs consistently, keep the model loaded to avoid repeated initialization costs, return confidence information, and maintain adequate performance under concurrent requests.

The Flask web service provides these functions through the analysis pipeline. Uploaded images are converted to RGB, resized to 224×224, normalized, and passed to the loaded model. The service then returns a Real/Fake prediction with a confidence score. Fig. 2 shows the modular relationship among the user interface, Flask API, data pipeline, and EfficientNetB0 inference module.

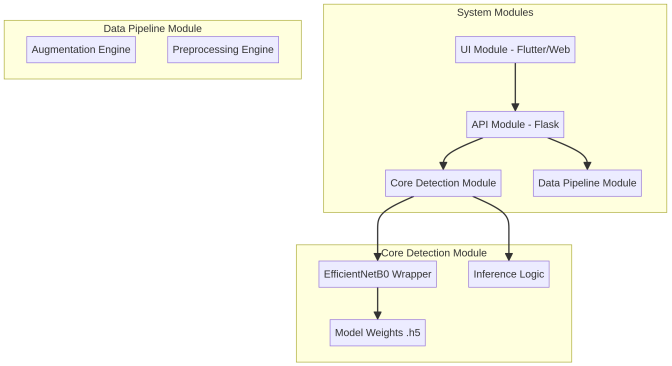


Fig. 2. Modular implementation linking the user interface, Flask API, data pipeline, and EfficientNetB0 inference module.

VI. Conclusion

This paper presents a deployable DeepFake image detector based on EfficientNetB0 and transfer learning. The experimental setup used the FaceForensics++ and Celeb-DF datasets with a common image resolution of 224×224 RGB pixels. Model development included augmentation that mimics real-world degradation, a lightweight classifier head, two-stage training, and multiple forms of evaluation. The detector achieved 92.8% validation accuracy, 92.5% precision, 93.1% recall, and a 92.8% F1-score based on the confusion matrix in Table III. Under dataset shift from FaceForensics++ to Celeb-DF, cross-dataset accuracy decreased to 84.2%. Robustness testing produced 89.5% accuracy at JPEG quality factor 70. Thus, the detector remained effective under moderate compression, while dataset shift caused a larger reduction in performance.

The work also demonstrates deployability through a Flask RESTful service connected to a web client and a Flutter mobile client. This architecture supports real-time verification scenarios in which the system must be fast, reliable, accessible, and modular. Generalizability remains the main research challenge. Future work should therefore evaluate newer generation techniques, apply strictly documented source-level splits, report additional measures such as area under the receiver operating characteristic curve and calibration, and extend detection to video and multimodal content.

References

[1] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 1–11, doi: 10.1109/ICCV.2019.00009.

[2] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3207–3216.

[3] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[5] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.

[6] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[7] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. 15th IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.

[8] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[9] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2307–2311.

[10] L. Verdoliva, "Media forensics and DeepFakes: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020, doi: 10.1109/JSTSP.2020.3002101.

[11] I. Altalahin, S. AlZu'bi, A. Alqudah, and A. Mughaid, "Unmasking the truth: A deep learning approach to detecting deepfake audio through MFCC features," in *2023 International Conference on Information Technology (ICIT)*, 2023, pp. 511–518, doi: 10.1109/ICIT58056.2023.10226172.

[12] M. Farfoura, M. A. Alia, I. Mashal, and A. Hnaif, "Arabic fake news detection using deep neural network transformers," in *2024 International Jordanian Cybersecurity Conference (IJCC)*, 2024, doi: 10.1109/IJCC64742.2024.10847275.

[13] T. G. Kanaan and M. F. Hamza, "AI-driven fake news detection using multimodal features and hybrid deep learning models," *International Journal of Advances in Soft Computing and its Applications*, vol. 18, no. 1, pp. 382–395, 2026, doi: 10.15849/ijasca.v18i1.51.

[14] E. A. Mohammed and A. Lakizadeh, "Benchmarking vision transformers for satellite image classification based on data augmentation techniques," *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 1, pp. 98–114, 2025, doi: 10.15849/IJASCA.250330.06.

[15] E. A. G. Aldhaher and A. Lakizadeh, "IRU-Net: Brain tumor detection in MRI images using the end-to-end inception residual UNet model," *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 2, pp. 83–100, 2025, doi: 10.15849/IJASCA.250730.05.

[16] S. Azfar, A. Nadeem, K. Ahsan, A. Mehmood, A. Ali, S. S. Al Qahtany, and H. Almoamari, "Automated system for detecting, identifying, and preventing cotton leaf and boll diseases using deep learning," *International Journal of Advances in Soft Computing and its Applications*, vol. 17, no. 1, pp. 67–97, 2025, doi: 10.15849/IJASCA.250330.05.

N-DQN-RRT: A Hybrid Deep Q-Network Guided RRT Algorithm for an Efficient Path Planning in Two-Dimensional Environments

Wai Yan Naing Win[1], K. Kanagalakshmi[2]*, K. Priyadharshini[3],
Zeyad A. T Ahmed[4], R. RajiniGanth[5], P. Ganesh Kumar[6], Atif Mahmood[7]*
Faculty of Data Science and Information Technology,
INTI International University, Nilai, 71800, Negeri Sembilan, Malaysia[1,4]
School of Science and Computer Studies, CMR University, Bangalore, Karnataka, India[2]
Jeppiaar Engineering College, Affiliated with Anna University, Chennai, Tamil Nadu, India[3]
SNS College of Engineering, Department of Computer Science and Technology, Coimbatore, India[5]
Department of Computer Science and Engineering, College of Engineering, Anna University, Chennai, Tamil Nadu, India[6]
Department of Computer System and Technology-Faculty of Computer Science and Information Technology,
Universiti Malaya, Kuala Lumpur, Malaysia[7]

***Abstract*—Autonomous navigation requires path-planning algorithms that can find safe and efficient routes while avoiding obstacles. Classical sampling-based planners such as Rapidly-exploring Random Tree (RRT) and RRT* are widely used because they are flexible and easy to apply in different environments. However, these methods often produce longer paths and require many samples in complex spaces. This paper proposes Neural Deep-Q Network RRT (N-DQN-RRT), a hybrid path-planning algorithm that integrates a Deep Q-Network (DQN) agent into an RRT*-style planning framework. The proposed method uses a directed-random sampling strategy. In this strategy, 75% of samples are guided by the trained DQN policy, 5% are assigned to goal bias, and 20% are sampled uniformly at random. The random component is retained to reduce premature policy bias and maintain exploration. The planner also uses an adaptive rewiring radius that changes according to local obstacle conditions. This allows the method to improve local path quality while avoiding unnecessary neighbor searches in cluttered regions. The algorithm was evaluated in three static two-dimensional environments with increasing map sizes of 10×10, 25×25, and 50×50. Each environment used randomized obstacle layouts and 100 independent trials, giving 300 trials per algorithm. The proposed method was compared with RRT and RRT* using six metrics: path length, steering angle, iteration count, sampling size, computation time, and success rate. Results were reported as mean $\pm$ standard deviation with 95% confidence intervals. N-DQN-RRT achieved a 100% success rate across all 300 trials, while RRT achieved 10–100% and RRT* achieved 14–100%, depending on map complexity. In the largest environment, N-DQN-RRT reduced path length by 11.9%, steering angle by 49.5%, sampling size by 75.1%, and computation time by 91.9% compared with RRT*. These results show that DQN-guided sampling can improve the empirical efficiency of RRT*-style planning in static two-dimensional obstacle environments.**

***Keywords*—Path planning; Rapidly-Exploring Random Trees (RRT); RRT*; Deep Q-Network (DQN); deep reinforcement learning; resilient infrastructure**

I. Introduction

Deep learning has been successfully applied across a wide range of intelligent perception and decision-making tasks, demonstrating its capability to learn complex representations from data [1]. Autonomous navigation is an important part of modern mobile robotics. A mobile robot must move from a start position to a target position while avoiding obstacles. This task is known as path planning. It is still a difficult problem because the planner must find a safe path under limited time and computational resources [2], [3].

The planning environment strongly affects the choice of method. In a static environment, obstacles remain fixed during planning. The robot can plan using a known map of the configuration space. In a dynamic environment, obstacles may move or appear during operation. This makes planning more difficult because the robot may need to update or replan its route during execution [3], [2]. For this reason, efficient and reliable path planning remains a key requirement for autonomous robots.

Classical graph-search methods such as A*, Dijkstra, and breadth-first search use a discretized representation of the environment. These methods are effective when the search space is small and the map resolution is manageable. However, their memory use and search cost increase as the map becomes larger or more detailed. This makes them less practical for large planning spaces and high-resolution maps [4], [2].

Sampling-based planners address this limitation in a different way. They do not need to search the full configuration space explicitly. Instead, they build a roadmap or tree by sampling points from the free space and connecting them through local collision checks [5], [2]. Rapidly-exploring Random Tree, or RRT, is one of the most common sampling-based planners. It grows a tree from the start position by repeatedly sampling the environment and extending the tree toward the sampled point. RRT is useful because it can find feasible paths in complex spaces.

*Corresponding author.

However, its paths are often irregular and longer than necessary. Its performance also depends strongly on random sampling [4], [2].

Several RRT variants have been proposed to improve path quality and convergence. RRT* adds parent selection and rewiring to improve path cost over time. Bi-directional RRT* grows two trees from the start and goal regions. Informed RRT* limits sampling after an initial solution is found so that search is focused on a more useful region of the space [6], [7]. These methods improve classical RRT in many cases. Still, they rely heavily on hand-designed sampling rules. They do not learn from previous planning experience.

Learning-based planning has been studied as a way to guide sampling toward more promising regions. Neural and generative models can learn useful structure from maps or previous paths. These learned priors can reduce unnecessary exploration and improve planning efficiency [8], [9]. Deep reinforcement learning also offers a useful framework for navigation. An agent can learn a policy through rewards and penalties. This allows the planner to prefer actions that move toward the goal and avoid obstacles [10], [11], [12].

Deep Q-Network, or DQN, is a value-based reinforcement learning method. It is suitable when the action space can be represented by a finite set of actions. In path planning, this makes DQN useful for selecting movement directions or guiding sample generation. However, a learned policy alone may be biased by its training distribution. It may also fail when the test environment differs from the training maps. For this reason, a hybrid planner can be useful. It can combine learned guidance with a random sampling component.

This paper proposes N-DQN-RRT, a hybrid path-planning algorithm for static two-dimensional environments. The method integrates a trained DQN agent into an RRT*-style planning framework. The DQN guides most sampling decisions toward promising directions. A random sampling component is retained to preserve exploration behavior and reduce policy bias. The planner also uses an adaptive rewiring radius. The radius is reduced near cluttered regions and expanded in more open regions. This allows the planner to improve path quality while reducing unnecessary neighbor searches.

The main contribution of this work is an empirical evaluation of the proposed N-DQN-RRT planner. The algorithm is tested in randomized two-dimensional obstacle environments with increasing size and complexity. Its performance is compared with classical RRT and RRT*. The evaluation uses path length, steering angle, iteration count, sampling size, computation time, and success rate. The goal is not to prove formal optimality or probabilistic completeness. Instead, the goal is to show whether DQN-guided sampling and adaptive rewiring can improve practical planning performance under the tested simulation conditions.

The contributions of this paper are summarized as follows:

- A hybrid DQN-guided RRT* planner is proposed for static two-dimensional path planning.
- The planner uses a seven-feature state representation that includes goal direction, obstacle proximity, and local free-space information.
- An 80/20 guided-random sampling strategy is used to balance learned guidance and stochastic exploration.
- An adaptive rewiring radius is introduced to adjust local optimization according to obstacle conditions.
- The method is evaluated across three randomized map sizes and compared with RRT and RRT* using six performance metrics.

II. Related Work

Path planning has been studied through classical search methods, sampling-based planners, learning-guided planners, and reinforcement learning approaches. This section reviews these groups and shows the gap addressed by the proposed N-DQN-RRT method (Table I).

A. Classical Sampling-Based Path Planning

Sampling-based planners are widely used in mobile robot path planning. They are useful when the configuration space is large or difficult to describe fully. Instead of building a complete grid of the space, these methods sample points from the free region and connect them through local collision checks [13], [2].

Rapidly-exploring Random Tree, or RRT, is one of the most common sampling-based planners. It grows a tree from the start point toward randomly sampled points. This makes RRT suitable for single-query planning problems. It can often find a feasible route in complex spaces. However, the path is usually not smooth or short. The result also depends strongly on the random samples [14], [4].

RRT* improves RRT by adding parent selection and rewiring. These steps reduce the path cost as the tree grows. Still, RRT* may require many samples before it converges to a high-quality path. Other variants improve the search process in different ways. Bi-directional RRT* grows two trees from the start and goal regions. Informed RRT* limits the sampling region after a first solution is found. Hybrid RRT* methods also combine global sampling with local improvement rules [6], [7]. These methods improve classical RRT, but they still depend on fixed sampling rules. They do not learn from earlier planning experience.

B. Learning-Guided Sampling-Based Planning

Learning-based planning tries to reduce unnecessary exploration. The main idea is simple. A model learns where useful samples are likely to appear. The planner can then spend more effort in promising regions.

Generative models have been used for this purpose. For example, GAN-based planners can use an environment image to predict a region where a feasible path is likely to exist. This region is then used as a sampling guide for a classical planner [8]. Such methods can improve convergence in tested environments. However, their performance depends on the learned map representation and the similarity between training and test environments.

Neural sampling methods have also been used with RRT-style planners. Neural RRT* uses a convolutional neural

TABLE I. SUMMARY OF RELATED WORK ON SAMPLING-BASED, LEARNING-BASED, AND DRL-BASED PATH PLANNING

Research group	Representative methods	Main idea	Key limitations	Related references
Classical sampling-based planning	RRT, RRT*, PRM	Builds a tree or roadmap through random samples in the free space. These methods are useful when the full configuration space is difficult to search directly.	Path quality can be poor. Convergence may be slow. Performance depends strongly on random samples.	[14], [4], [2]
Improved RRT variants	Bi-RRT*, Informed RRT*, hybrid RRT*	Improves classical RRT by using rewiring, bidirectional search, or restricted sampling after an initial path is found. These methods can improve path cost and search efficiency.	Still depends on hand-designed sampling rules. It does not learn from previous planning experience. Performance may degrade in cluttered or narrow regions.	[6], [7]
Learning-guided sampling	GAN-RRT, Neural RRT*, DNN-guided sampling	Uses neural or generative models to guide samples toward promising regions of the map. This can reduce unnecessary exploration and improve planning efficiency.	Requires training data. The learned sampler may be sensitive to changes between training and test environments. Generalization remains a concern.	[8], [9], [15]
Deep reinforcement learning planning	DQN, PPO, SAC, TD3	Learns navigation behavior from interaction with the environment. The reward function encourages goal progress and penalizes unsafe actions.	Training can be expensive. Results depend on reward design and hyperparameters. Some methods have difficulty transferring from simulation to real robots.	[10], [16], [17], [18], [19], [20]
Hybrid RL and RRT planning	RL-RRT, DRL-RRT, RRT*-RL	Combines learned guidance with the tree structure of RRT or RRT*. The planner keeps classical expansion behavior while using learning to reduce inefficient exploration.	May lose formal planning guarantees if learning dominates sampling. A learned policy can be biased by the training distribution. Ablation is needed to show which component gives the improvement.	[11], [12], [21]
Proposed direction	N-DQN-RRT	Uses a DQN-guided sampler inside an RRT*-style framework. It keeps random sampling for exploration and adds an adaptive rewiring radius for local path improvement.	Current evaluation is limited to static two-dimensional environments. Dynamic scenes, cross-map transfer, and hardware validation remain future work.	This study

network to guide non-uniform sampling for optimal path planning [9]. Deep neural network enhanced sampling has also been studied in three-dimensional planning spaces [15]. These studies show that learned priors can improve sample efficiency. At the same time, they also show a common limitation. A learned sampler may lose reliability when the test environment differs from the training data.

C. Deep Reinforcement Learning for Robot Navigation

Deep reinforcement learning has become a common tool for robot navigation. In this setting, an agent learns from interaction with the environment. The reward function encourages goal progress and penalizes unsafe behavior. This makes DRL useful for navigation tasks where direct planning is difficult [10], [16].

Value-based methods such as Deep Q-Network are useful when the action space is discrete. A DQN can learn the value of each possible action from the current state. This is suitable for planners that choose among a fixed set of movement directions. Improved DQN variants have been studied for mobile robot path planning, often by changing the reward function or network structure [17], [22]. However, DQN methods can be sensitive to reward design. They may also suffer when the learned policy is applied outside the training distribution.

Policy-gradient and actor-critic methods are also common in navigation. Methods such as PPO, SAC, and TD3 have been applied to dynamic path planning and obstacle avoidance [18], [19], [20]. These methods can handle more complex action spaces. They can also learn flexible behavior in dynamic scenes. Their main drawbacks are training cost, hyperparameter sensitivity, and the gap between simulation and real deployment.

D. Hybrid Planning with Reinforcement Learning and RRT

A different line of work combines learning with classical sampling-based planners. This is useful because the planner keeps the tree structure of RRT while the learning model guides part of the search. RL-RRT is one example. It uses a learned policy and a reachability estimator to guide tree growth in kinodynamic planning [11]. Other DRL-integrated RRT methods use learned policies to improve sample selection or path expansion [12]. Recent work has also explored reinforcement learning inside RRT*-style frameworks for mobile robot planning [21].

These hybrid methods are important because they do not fully replace classical planning. Instead, they add learning where random exploration is inefficient. This helps preserve the structure of sampling-based planning while adding experience-based guidance. However, many existing studies use different environments, different learning models, and different evaluation settings. This makes direct comparison difficult.

E. Research Gap

The reviewed studies show that classical RRT methods are flexible but sample inefficient. They also show that learned sampling can improve planning speed and path quality. However, there is still a need for a simple value-based hybrid planner that keeps the RRT* structure and uses DQN guidance in a controlled way.

TABLE II. PLANNER PARAMETERS FOR THE THREE MAP SETTINGS

Parameter	Map 1	Map 2	Map 3
Environment size	10×10	25×25	50×50
Step size η	1.0	1.0	1.0
Maximum iterations	250	500	1500
Goal radius	1.0	1.0	1.0
Start position	$(1, 1)$	$(1, 1)$	$(1, 1)$
Goal position	$(9, 9)$	$(24, 24)$	$(49, 49)$
Base rewire radius	2.0	2.0	2.0

This paper addresses that gap through N-DQN-RRT. The method uses a DQN-guided sampler inside an RRT*-style framework. It also keeps a random sampling component to reduce policy bias. An adaptive rewiring radius is added to improve local path quality. The method is evaluated in static two-dimensional random obstacle environments with increasing map size. The goal is to test whether learned guidance can improve practical planning performance when compared with classical RRT and RRT*.

III. METHODOLOGY

This section describes the experimental setup and the proposed N-DQN-RRT algorithm. The study uses a simulation-based design. The proposed method is compared with classical RRT and RRT*. All planners are evaluated under the same map settings. They also use the same start and goal positions.

A. Experimental Framework

The experiments use a quantitative simulation framework. Three algorithms are evaluated. These are RRT, RRT*, and the proposed N-DQN-RRT. The goal is to compare their planning performance in static two-dimensional environments.

All algorithms are implemented in Python. NumPy is used for numerical operations. Matplotlib is used for visualization. PyTorch is used to build and train the DQN model. SciPy is used for statistical analysis. The experiments are run on a Google Colab CPU environment.

Each map configuration is tested using 100 independent trials. No fixed seed is used during the main trials. This keeps the natural stochastic behavior of each planner. The repeated trials are then used to report the mean, standard deviation, and confidence interval of each metric.

B. Map Configuration

Three square environments are used. The map sizes are 10×10, 25×25, and 50×50. The start point is placed near the lower-left corner. The goal point is placed near the upper-right corner. Each trial uses a randomly generated obstacle layout.

The obstacles include circles, rectangles, squares, and triangles. The number of obstacles increases with map size. This makes the larger maps more difficult. All obstacles remain fixed during one planning run. Therefore, the experiments are limited to static environments (Table II).

C. Collision Checking

Collision checking is applied to both nodes and edges. A sampled node is rejected if it lies inside an obstacle or too close to an obstacle boundary. An edge is also rejected if any interpolated point along the edge intersects an obstacle. This prevents the planner from accepting paths that pass through occupied space.

For each edge, points are sampled at a fixed interpolation interval. If any point violates the collision rule, the full edge is discarded. A small collision buffer is used to keep the path away from obstacle boundaries.

D. Baseline Planners

RRT grows a tree from the start point. At each iteration, it samples a point from the map. The nearest tree node is found. The tree is then extended from the nearest node toward the sampled point.

Let q_{near} be the nearest node and q_{sample} be the sampled point. The new node is computed as:

$$q_{\text{new}} = q_{\text{near}} + \min\left(\eta, \left\|q_{\text{sample}} - q_{\text{near}}\right\|_2\right) \frac{q_{\text{sample}} - q_{\text{near}}}{\left\|q_{\text{sample}} - q_{\text{near}}\right\|_2}. \quad (1)$$

Here, η is the step size.

The Euclidean distance between two points $q_i = (x_i, y_i)$ and $q_j = (x_j, y_j)$ is:

$$d(q_i, q_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (2)$$

RRT* extends RRT by adding parent selection and rewiring. For each new node, the algorithm searches nearby nodes. It selects the parent that gives the lowest collision-free cost. It then checks whether nearby nodes can be rewired through the new node to reduce path cost. This improves path quality over time.

E. Proposed N-DQN-RRT Algorithm

The proposed N-DQN-RRT algorithm combines DQN-guided sampling with an RRT*-style tree structure. The method has two phases. The first phase trains a DQN agent. The second phase uses the trained agent to guide sampling during planning.

The planner does not rely only on the DQN. It also keeps random sampling. This reduces the risk of policy bias. It also helps the tree recover when the learned policy gives a poor direction.

1) State representation: The DQN agent receives a seven-dimensional state vector. The state describes the current node, the goal, the nearest obstacle, and the local free space. It is defined as

$$s_t = \left[d_x,\, d_y,\, d_{\text{goal}},\, d_{\text{obs}},\, \delta x_{\text{obs}},\, \delta y_{\text{obs}},\, f_{\text{free}}\right]^T. \quad (3)$$

The terms d_x and d_y are the normalized horizontal and vertical offsets from the current node to the goal. The term d_{goal} is the normalized distance to the goal. The term d_{obs}

is the normalized distance to the nearest obstacle. The terms δx_{obs} and δy_{obs} describe the relative direction of the nearest obstacle. The term f_{free} is the fraction of nearby probe directions that are collision-free.

2) Action space: The action space contains eight discrete directions. The directions are spaced by 45°. Each action selects a direction for guided sampling. The sample distance is drawn from a small random range. This keeps the guided sampler from producing the same path repeatedly.

3) Q-network architecture: The DQN uses a feed-forward neural network. The input layer has seven units. There are two hidden layers with 128 units each. ReLU activation is used in the hidden layers. The output layer has eight units. Each output gives the estimated Q-value of one action. A target network is used during training. The target network has the same architecture as the main Q-network. Its weights are copied from the main network at fixed intervals. This makes the target values more stable during learning.

4) Reward function: The reward function encourages three behaviors. The agent should move toward the goal. It should avoid obstacles. It should avoid unnecessary long movement.

The reward at step t is defined as:

$$r_t = \begin{cases} -10, & \text{if a collision occurs,} \\ +100, & \text{if } d(q_{t+1}, q_{\text{goal}}) \le r_{\text{goal}}, \\ 3\Delta d_{\text{goal}} + 5\Delta c_{\text{obs}} - 0.05, & \text{otherwise.} \end{cases} \tag{4}$$

Here, r_{goal} is the reward threshold near the goal. It is different from the planning goal radius used to stop the algorithm. The term Δd_{goal} is the reduction in distance to the goal:

$$\Delta d_{\text{goal}} = d(q_t, q_{\text{goal}}) - d(q_{t+1}, q_{\text{goal}}). \tag{5}$$

The term Δc_{obs} is the change in obstacle clearance:

$$\Delta c_{\text{obs}} = c(q_{t+1}) - c(q_t). \tag{6}$$

A positive value means that the new position is farther from the nearest obstacle. A negative value means that it is closer to an obstacle.

5) DQN training: The agent is trained with experience replay. Each transition is stored as:

$$(s_t, a_t, r_t, s_{t+1}, done_t). \tag{7}$$

Mini-batches are sampled from the replay buffer during training. The Q-network is trained by minimizing the mean squared Bellman error:

$$L(\theta) = \mathbb{E}\left[\left(y_t - Q(s_t, a_t; \theta)\right)^2\right]. \tag{8}$$

The target value is

$$y_t = r_t + \gamma(1 - done_t) \max_{a'} Q(s_{t+1}, a'; \theta^-). \tag{9}$$

Here, θ denotes the main network weights. The term θ^- denotes the target network weights. The discount factor is γ. The term $done_t$ is 1 when the episode ends and 0 otherwise.

The agent uses an epsilon-greedy policy during training. At the beginning, the agent explores more. As training continues, exploration is reduced. The epsilon value is updated as:

$$\epsilon_{k+1} = \max(\epsilon_{\min}, \beta\epsilon_k). \tag{10}$$

Here, $\epsilon_{\min}$ is the minimum exploration rate. The term β is the decay factor.

6) Adaptive rewiring radius: N-DQN-RRT uses an adaptive rewiring radius. The radius changes according to local obstacle information. It becomes smaller near cluttered regions. It becomes larger in open regions.

The adaptive radius is computed as:

$$r_{\text{adaptive}} = \text{clip}\left(r_{\text{base}}\left(0.5 + 0.8d_{\text{obs}} + 0.7f_{\text{free}}\right), 0.6r_{\text{base}}, 2.0r_{\text{base}}\right). \tag{11}$$

Here, r_{base} is the base rewiring radius. The term d_{obs} is the normalized obstacle distance. The term f_{free} is the local free-space fraction. The function $\text{clip}(\cdot)$ keeps the radius within the lower and upper bounds.

7) Planning phase: During planning, the tree starts at the start node. At each iteration, a candidate sample is generated in one of three ways. A small portion of samples is placed at the goal. Another portion is sampled uniformly at random. The remaining samples are generated through the trained DQN policy.

The sampling design follows an 80/20 directed-random strategy. In this design, 75% of samples are generated by the DQN policy and 5% are placed at the goal as goal bias. The remaining 20% are sampled uniformly at random. Thus, 80% of the samples are directed and 20% preserve global exploration.

The sampling rule is:

$$q_{\text{sample}} = \begin{cases} q_{\text{goal}}, & \text{with probability } 0.05, \\ q_{\text{rand}}, & \text{with probability } 0.20, \\ q_{\text{DQN}}, & \text{with probability } 0.75. \end{cases} \tag{12}$$

After a sample is generated, the nearest node is found. A new node is created using Eq. (1). The node is added only if the point and edge are collision-free. The best parent is selected from nearby nodes. Then nearby nodes are rewired if the new connection lowers their path cost.

The planning process stops when the tree reaches the goal region. The final path is extracted by tracing parent links from the goal node back to the root.

F. Hyperparameter Settings

The training and planning hyperparameters of N-DQN-RRT are presented in Table III.

TABLE III. N-DQN-RRT HYPERPARAMETERS

Hyperparameter	Value
Training episodes	400
Maximum steps per episode	300
Batch size	64
Learning rate	0.001
Optimizer	Adam
Discount factor γ	0.95
Initial epsilon ϵ_0	1.0
Epsilon decay β	0.997
Minimum epsilon during training	0.05
Planning epsilon	0.02
Replay buffer capacity	8000
Hidden layer size	128×128
Network structure	$7 \rightarrow 128 \rightarrow 128 \rightarrow 8$
Target update frequency	50 gradient steps
Collision buffer	0.05 units
Edge interpolation step	0.02 units
Base rewiring radius	2.0 units

G. Algorithm Summary

Algorithm 1 presents the pseudocode of the proposed N-DQN-RRT algorithm. Fig. 1 shows the corresponding flowchart. The algorithm starts from the root node. It then expands the tree until the goal region is reached or the maximum number of iterations is exceeded.

Algorithm 1 N-DQN-RRT

Require: Start node q_{start}, goal node q_{goal}, obstacle set, maximum iterations, trained DQN agent
Ensure: Collision-free path or failure

```
Initialize tree T with root node q_start
for i = 1 to maximum iterations do
    Generate q_sample using Eq. (12)
    Find nearest node q_near in T
    Compute q_new using Eq. (1)
    if q_new is in collision or edge (q_near, q_new) is in collision then
        Continue
    end if
    Compute r_adaptive using Eq. (11)
    Find neighbor nodes within r_adaptive
    Select the lowest-cost collision-free parent
    Add q_new to T
    Rewire nearby nodes if the new connection reduces cost
    if d(q_new, q_goal) is within the goal radius then
        Add goal node
        return extracted path
    end if
end for
return failure
```

Fig. 1 shows the main steps of N-DQN-RRT. The tree is first initialized at the start node. At each iteration, a candidate sample is generated using goal bias, random sampling, or the trained DQN policy. The candidate node and edge are then checked for collision. If the candidate is valid, the planner selects a parent, adds the new node, and applies adaptive rewiring. The process stops when the goal region is reached. If the maximum number of iterations is reached first, the planner returns failure.

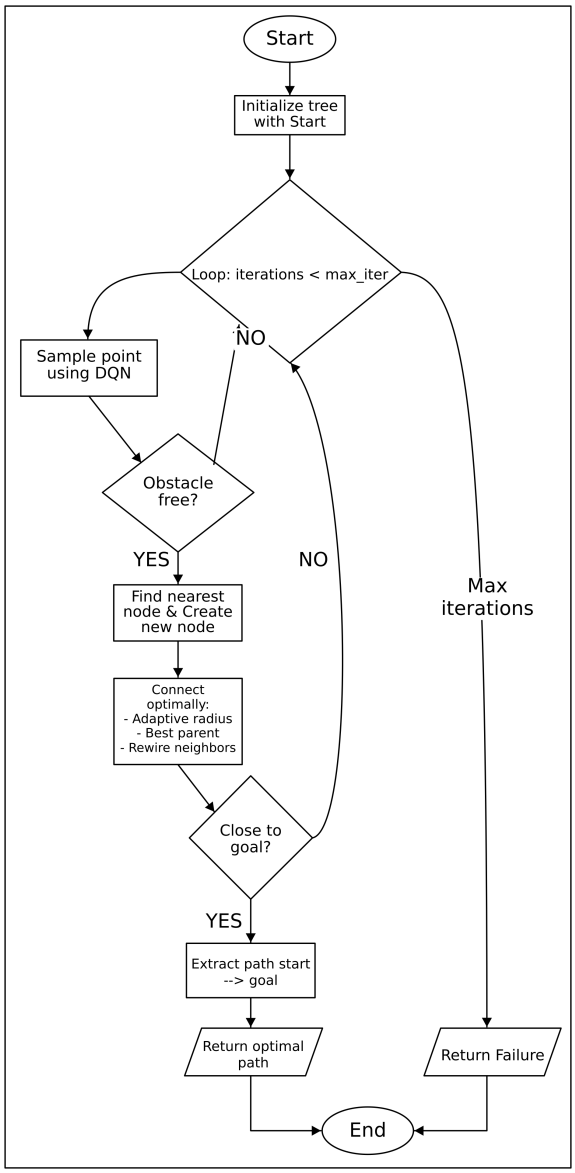


Fig. 1. Flowchart of the proposed N-DQN-RRT path planning algorithm.

H. Evaluation Metrics

Six metrics are used to evaluate the planners. Path length measures the total Euclidean length of the final path. Steering angle measures the average turning angle between consecutive path segments. Iterations measure the number of tree expansion steps. Sampling size measures the number of nodes in the final tree. Computation time measures wall-clock planning time. Success rate measures the percentage of trials that reach the goal within the iteration limit.

For each metric, the mean and standard deviation are

reported across 100 trials. A 95% confidence interval is also computed for the main continuous metrics. Path length is calculated only for successful trials. Success rate is reported over all trials.

The 95% confidence interval is computed as

$$CI = \bar{x} \pm t_{\alpha/2,n-1}\frac{s}{\sqrt{n}}, \tag{13}$$

where, $\bar{x}$ is the sample mean. The term s is the sample standard deviation. The term n is the number of trials. The term $t_{\alpha/2,n-1}$ is the critical value from the t distribution.

I. Reproducibility Details

The implementation uses a three-layer Q-network with input size 7 and output size 8. All state features are normalized before being passed to the network. The optimizer is Adam with a learning rate of 0.001. The loss function is the mean squared Bellman error. The collision buffer is set to 0.05 units. The edge interpolation step is set to 0.02 units. The base rewiring radius is set to 2.0 units.

The main experiments use 100 independent trials per map. This supports statistical reporting under stochastic planning. For exact reproducibility, the final implementation should also report the obstacle generation rules, software versions, hardware configuration, and random seed list.

IV. Results and Discussion

This section presents the results of RRT, RRT*, and the proposed N-DQN-RRT. The algorithms were tested on three static two-dimensional maps. The map sizes were 10×10, 25×25, and 50×50. Each map setting was evaluated using 100 independent trials. The start and goal positions were fixed for each map. The obstacle layout was randomly generated in each trial.

Six metrics were used for comparison. These were path length, steering angle, iteration count, sampling size, computation time, and success rate. Path length, steering angle, iteration count, sampling size, and computation time were computed from successful trials. Success rate was computed over all 100 trials.

A. Map 1 Results: 10×10 Environment

The first experiment used a 10×10 environment. The start position was $(1.0, 1.0)$. The goal position was $(9.0, 9.0)$. The step size was 1.0. The maximum number of iterations was 250 for all algorithms (Table IV).

All three algorithms reached the goal in every trial. This shows that the smallest map was manageable for all planners under the given iteration limit. However, N-DQN-RRT produced better paths and used fewer samples.

Compared with RRT*, N-DQN-RRT reduced the mean path length from 13.90 m to 12.50 m. This is a reduction of about 10.1%. It also reduced the mean steering angle from 40.07° to 22.63°. This indicates a smoother path. The iteration count dropped from 79 to 24. The sampling size also dropped from 61 to 22.

The confidence intervals for path length do not overlap between N-DQN-RRT and RRT*. This supports a meaningful difference in the tested trials. It should not be treated as mathematical proof. It shows that the observed improvement is unlikely to be caused only by random variation in these experiments.

B. Map 2 Results: 25×25 Environment

The second experiment used a 25×25 environment. The start position was $(1.0, 1.0)$. The goal position was $(24.0, 24.0)$. The step size was 1.0. The maximum number of iterations was 500 (Table V).

The second map shows a clear difference between the planners. RRT reached the goal in 28% of the trials. RRT* reached the goal in 38% of the trials. N-DQN-RRT reached the goal in all trials. This is an increase of 62 percentage points over RRT*.

Among successful trials, N-DQN-RRT also produced shorter paths. Its mean path length was 36.60 m. The mean path length of RRT* was 41.10 m. This is a reduction of about 10.9%. The mean steering angle was also much lower. This suggests that the learned sampling policy helped produce smoother routes.

N-DQN-RRT required fewer iterations and fewer nodes than RRT*. The iteration count decreased from 404 to 186. The sampling size decreased from 263 to 103. The computation time also decreased from 0.60 s to 0.11 s. This reduction is mainly due to the smaller tree size and fewer expansion steps.

The standard deviation of the N-DQN-RRT iteration count is still high. This shows that some random obstacle layouts were harder than others. Even so, the method reached the goal in every trial.

C. Map 3 Results: 50×50 Environment

The third experiment used the largest environment. The map size was 50×50. The start position was $(1.0, 1.0)$. The goal position was $(49.0, 49.0)$. The step size was 1.0. The maximum number of iterations was 1500 (Table VI).

The 50×50 map was the most difficult case. RRT succeeded in 10% of the trials. RRT* succeeded in 14% of the trials. N-DQN-RRT succeeded in all 100 trials. This is an increase of 86 percentage points over RRT*.

The path length also improved. N-DQN-RRT achieved a mean path length of 76.80 m. RRT* achieved 87.20 m. This is a reduction of about 11.9%. The steering angle was reduced from 33.76° to 17.04°. This again shows smoother path behavior.

The largest gain was seen in computation time and sampling size. N-DQN-RRT used 238 nodes on average. RRT* used 955 nodes on average. The computation time decreased from 6.39 s to 0.52 s. This suggests that the DQN-guided sampler reduced unnecessary tree growth in the tested maps.

The standard deviation of N-DQN-RRT computation time is high. This means that a few trials were much harder than the average case. The method still reached the goal in every run.

TABLE IV. MAP 1 RESULTS OVER 100 TRIALS IN THE 10×10 ENVIRONMENT

Metric	RRT	RRT*	N-DQN-RRT
Path length	14.60 ± 1.40 m	13.90 ± 1.06 m	$\mathbf{12.50 \pm 0.83}$ m
Iterations	69 ± 24	79 ± 40	$\mathbf{24 \pm 10}$
Sampling size	55 ± 17	61 ± 26	$\mathbf{22 \pm 5}$
Steering angle	$42.67^\circ \pm 8.74^\circ$	$40.07^\circ \pm 8.26^\circ$	$\mathbf{22.63^\circ \pm 7.83^\circ}$
Computation time	0.02 ± 0.01 s	0.04 ± 0.05 s	$\mathbf{0.01 \pm 0.004}$ s
Success rate	100.0%	100.0%	**100.0%**
95% CI for path length	[14.34, 14.89]	[13.68, 14.10]	$[\mathbf{12.32}, \mathbf{12.65}]$
95% CI for computation time	[0.0141, 0.0188]	[0.0300, 0.0500]	$[\mathbf{0.0074}, \mathbf{0.0087}]$

TABLE V. MAP 2 RESULTS OVER 100 TRIALS IN THE 25×25 ENVIRONMENT

Metric	RRT	RRT*	N-DQN-RRT
Path length	42.10 ± 2.83 m	41.10 ± 2.14 m	$\mathbf{36.60 \pm 1.87}$ m
Iterations	407 ± 67	404 ± 62	$\mathbf{186 \pm 95}$
Sampling size	270 ± 46	263 ± 44	$\mathbf{103 \pm 34}$
Steering angle	$36.79^\circ \pm 4.91^\circ$	$37.64^\circ \pm 4.38^\circ$	$\mathbf{17.82^\circ \pm 3.43^\circ}$
Computation time	0.34 ± 0.12 s	0.60 ± 0.27 s	$\mathbf{0.11 \pm 0.06}$ s
Success rate	28.0%	38.0%	**100.0%**
95% CI for path length	[41.05, 43.24]	[40.39, 41.79]	$[\mathbf{36.18}, \mathbf{36.93}]$
95% CI for computation time	[0.2938, 0.3891]	[0.5173, 0.6925]	$[\mathbf{0.0962}, \mathbf{0.1194}]$

TABLE VI. MAP 3 RESULTS OVER 100 TRIALS IN THE 50×50 ENVIRONMENT

Metric	RRT	RRT*	N-DQN-RRT
Path length	89.70 ± 3.41 m	87.20 ± 3.95 m	$\mathbf{76.80 \pm 3.11}$ m
Iterations	1259 ± 232	1345 ± 109	$\mathbf{317 \pm 203}$
Sampling size	901 ± 145	955 ± 86	$\mathbf{238 \pm 135}$
Steering angle	$35.92^\circ \pm 3.64^\circ$	$33.76^\circ \pm 2.94^\circ$	$\mathbf{17.04^\circ \pm 2.24^\circ}$
Computation time	3.56 ± 1.30 s	6.39 ± 1.33 s	$\mathbf{0.52 \pm 0.46}$ s
Success rate	10.0%	14.0%	**100.0%**
95% CI for path length	[87.23, 92.10]	[84.88, 89.45]	$[\mathbf{76.21}, \mathbf{77.45}]$
95% CI for computation time	[2.6353, 4.4926]	[5.6176, 7.1531]	$[\mathbf{0.4273}, \mathbf{0.6085}]$

This supports its practical stability under the tested random obstacle settings.

It is important to interpret the results carefully. For Maps 2 and 3, RRT and RRT* failed in many trials. Their path length and computation time values are based only on successful trials. These successful cases may represent easier obstacle layouts. Therefore, success rate is the most important metric for larger maps.

D. Cross-Environment Analysis

Table VII summarizes the performance of N-DQN-RRT across the three map sizes. The method kept a 100% success rate in all tested environments. The path length increased as the map size increased. This is expected because the start and goal were farther apart in larger maps.

The steering angle decreased from Map 1 to Map 3. This may be because larger maps provide more space for gradual movement. It may also reflect the effect of DQN-guided directional sampling. Further testing is needed before making a stronger claim.

The iteration count and sampling size increased with map size. This is expected because larger maps require longer paths and more expansion steps. However, the increase was still lower than the growth seen in RRT* computation time. In these experiments, N-DQN-RRT increased from 0.01 s to 0.52 s. RRT* increased from 0.04 s to 6.39 s. This indicates better empirical scaling under the tested settings.

E. Discussion

The results show that the proposed N-DQN-RRT planner improves practical planning performance in the tested static two-dimensional environments. The improvement is most visible in the larger maps. In those maps, RRT and RRT* often failed to reach the goal within the iteration limit. N-DQN-RRT reached the goal in every trial.

The main reason is the sampling strategy. The DQN policy guides most samples toward useful directions. The random sampling component keeps some global exploration. This helps reduce the risk of relying only on the learned policy. The goal bias also helps the planner converge near the target region.

The adaptive rewiring radius also contributes to path quality. In open regions, a larger radius allows the planner to consider more possible parent nodes. Near obstacles, a smaller radius reduces unnecessary neighbor checks. This can reduce collision-heavy rewiring attempts. It can also help produce smoother local connections.

The results should not be overinterpreted. The experiments do not prove probabilistic completeness or asymptotic optimality for N-DQN-RRT. They only show empirical improvement under the tested simulation conditions. The training and testing

TABLE VII. N-DQN-RRT PERFORMANCE ACROSS MAP CONFIGURATIONS

Metric	**Map 1:** 10×10	**Map 2:** 25×25	**Map 3:** 50×50
Success rate	100%	100%	100%
Path length	12.50 ± 0.83 m	36.60 ± 1.87 m	76.80 ± 3.11 m
Steering angle	$22.63^\circ \pm 7.83^\circ$	$17.82^\circ \pm 3.43^\circ$	$17.04^\circ \pm 2.24^\circ$
Iterations	24 ± 10	186 ± 95	317 ± 203
Sampling size	22 ± 5	103 ± 34	238 ± 135
Computation time	0.010 ± 0.004 s	0.110 ± 0.060 s	0.520 ± 0.460 s

maps were generated from the same general distribution. This means that performance on different map types is still uncertain. Examples include narrow passages, maze-like maps, dynamic obstacles, and real robot environments.

There are also reproducibility limits. The trials were repeated 100 times, but no fixed random seed list was used. This supports statistical reporting, but it does not provide exact run-to-run reproduction. Future experiments should include fixed seed lists, obstacle-generation details, and code release. Additional baselines such as Informed RRT*, Bi-RRT*, and goal-biased RRT* would also make the comparison stronger.

Overall, the results support the use of DQN-guided sampling as a practical improvement for RRT*-style planning in static 2D environments. The strongest evidence is the increase in success rate and the reduction in sampling size on larger maps. The method is promising, but it should be tested further before being claimed as general or deployment-ready.

V. CONCLUSION

This paper proposed and evaluated N-DQN-RRT, a hybrid path-planning algorithm for static two-dimensional environments. The method integrates a trained Deep Q-Network agent into an RRT*-style planning framework. It uses a seven-dimensional state representation that includes goal direction, obstacle distance, obstacle direction, and local free-space information. It also uses an 80/20 directed-random sampling strategy. In this strategy, 75% of samples are guided by the DQN policy, 5% are assigned to goal bias, and 20% are sampled uniformly at random. An adaptive rewiring radius is also used to adjust the local optimization range according to obstacle conditions.

The proposed method was implemented in Python and tested in three static two-dimensional random obstacle environments. The map sizes were 10×10, 25×25, and 50×50. Each map setting was evaluated using 100 independent stochastic trials. The performance of N-DQN-RRT was compared with classical RRT and RRT* using path length, steering angle, iteration count, sampling size, computation time, and success rate.

The results show that N-DQN-RRT improved empirical planning performance under the tested conditions. It achieved a 100% success rate across all 300 trials. In comparison, RRT achieved 28–100% success and RRT* achieved 38–100% success depending on the map size. In the largest 50×50 environment, N-DQN-RRT reduced path length by 11.9%, steering angle by 49.5%, sampling size by 75.1%, and computation time by 91.9% compared with RRT*. The non-overlapping 95% confidence intervals support the observed differences in the tested trials.

The main contribution of this work is empirical and architectural. The paper does not claim to be the first study on DQN-RRT integration. Instead, it studies a specific hybrid design that combines DQN-guided sampling, goal bias, random exploration, and adaptive rewiring within an RRT*-style planner. The results suggest that learned sampling can reduce unnecessary exploration and improve practical planning efficiency in static two-dimensional random obstacle environments.

This study has several limitations. First, the evaluation is limited to static two-dimensional maps. Dynamic obstacles, moving agents, and real-world sensing uncertainty are not considered. Second, the DQN agent is trained and tested on maps generated from the same general distribution. This limits the strength of the generalization claim. Third, no physical robot experiment is included. The simulation does not model sensor noise, actuator delay, localization error, or communication latency.

Future work should extend N-DQN-RRT to dynamic environments that require online replanning. The method should also be tested on physical mobile robots using sensors such as LiDAR or depth cameras. Further experiments should include three-dimensional planning tasks for UAVs and robotic manipulators. Ablation studies are also needed to measure the contribution of each component. These studies should test different DQN-random sampling ratios, remove adaptive rewiring, and compare fixed-radius and adaptive-radius variants. Cross-map generalization tests should also be added by training on one obstacle distribution and testing on different map structures such as narrow passages and maze-like environments.

REFERENCES

[1] O. Al-Sanjary *et al.*, "A hybrid method of enhancing accuracy of facial recognition system using gabor filter and stacked sparse autoencoders deep neural network," *Applied Sciences*, vol. 12, no. 21, p. 11052, 2022.

[2] L. Yang *et al.*, "Path planning technique for mobile robots: A review," *Machines*, vol. 11, no. 10, p. 980, 2023.

[3] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, "A review on path planning and obstacle avoidance algorithms for autonomous mobile robots," *Journal of Robotics*, vol. 2022, p. 2538220, 2022.

[4] R. Agrawal, B. Singh, and R. Kumar, "Classical approaches for mobile robot path planning: A review," in *Proceedings of the 3rd IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2022, pp. 400–405.

[5] L. B. P. Nascimento, D. Barrios-Aranibar, V. G. Santos, D. S. Pereira, W. C. Ribeiro, and P. J. Alsina, "Safe path planning algorithms for mobile robots based on probabilistic foam," *Sensors*, vol. 21, no. 12, p. 4156, 2021.

[6] P. Xin, X. Wang, X. Liu, Y. Wang, Z. Zhai, and X. Ma, "Improved bidirectional rrt* algorithm for robot path planning," *Sensors*, vol. 23, no. 2, p. 1041, 2023.

[7] S. Ganesan, B. Ramalingam, and R. E. Mohan, "A hybrid sampling-based rrt* path planning algorithm for autonomous mobile robot navigation," *Expert Systems with Applications*, vol. 258, p. 125206, 2024.

[8] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative adversarial network based heuristics for sampling-based path planning," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 64–74, 2022.

[9] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural rrt*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[10] L. Xu and W. Zhang, "Survey on path planning based on deep reinforcement learning," in *Proceedings of the 2025 2nd International Conference on Machine Learning and Intelligent Computing*, ser. Proceedings of Machine Learning Research, vol. 278. PMLR, 2025, pp. 685–695. [Online]. Available: https://proceedings.mlr.press/v278/xu25a.html

[11] H. T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "Rl-rrt: Kinodynamic motion planning via learning reachability estimators from rl policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[12] H. Liu, Y. Gu, X. Li, and X. Xiao, "Deep reinforcement learning integrated rrt algorithm for path planning," in *Proceedings of the 2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, 2023, pp. 239–244.

[13] A. Alexander, K. Venkatesan, J. Mounsef, and K. Ramanujam, "A comprehensive survey of path planning algorithms for autonomous systems and mobile robots: Traditional and modern approaches," *IEEE Access*, 2025.

[14] A. Muhammad, N. R. H. Abdullah, M. A. H. Ali, I. H. Shanono, and R. Samad, "Simulation performance comparison of a*, gls, rrt and prm path planning algorithms," in *Proceedings of the 12th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, 2022, pp. 258–263.

[15] J. Wang, X. Jia, T. Zhang, N. Ma, and M. Q.-H. Meng, "Deep neural network enhanced sampling-based path planning in 3d space," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3434–3443, 2022.

[16] L. Yang, J. Bi, and H. Yuan, "Dynamic path planning for mobile robots with deep reinforcement learning," in *IFAC-PapersOnLine*, 2022, pp. 19–24.

[17] Y. Li, "An intelligent vehicle path planning algorithm based on an improved dqn for use in complex environments," in *Proceedings of the 7th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI)*, 2025, pp. 628–632.

[18] X. Jin and Z. Wang, "Proximal policy optimization based dynamic path planning algorithm for mobile robots," *Electronics Letters*, 2022.

[19] Y. Zhang and P. Chen, "Path planning of a mobile robot for a dynamic indoor environment based on an sac-lstm algorithm," *Sensors*, vol. 23, no. 24, p. 9802, 2023.

[20] P. Li, D. Chen, Y. Wang, L. Zhang, and S. Zhao, "Path planning of mobile robot based on improved td3 algorithm in dynamic environment," *Heliyon*, vol. 10, no. 11, p. e32167, 2024.

[21] H.-L. Pham, N.-N. Bui, and T.-V. Dang, "Hybrid path planning for wheeled mobile robot based on rrt-star algorithm and reinforcement learning method," *Journal of Robotics and Control (JRC)*, vol. 6, no. 4, pp. 2045–2051, 2025.

[22] X. Xu, Y. Cao, and X. Liu, "Improved dqn algorithm for path planning of autonomous mobile robots," 2023, research Square preprint.

Deep Learning and Computer Vision-Based Modeling for a Real-Time Traffic Congestion Monitoring Framework

Neil Dustin Benedict A. Agner, Angelo B. Dela Cruz,
Aster Benedict A. Mangabat, Raphael B. Alampay, Patricia Angela R. Abu
Ateneo Laboratory for Intelligent, Visual Environments, Ateneo de Manila University, Philippines

***Abstract*—As intelligent traffic systems evolve to manage complex urban mobility, conventional congestion estimation techniques, such as the time-windowed Volume-to-Capacity (V/C) ratio, fail to capture capture the real-time traffic situation. Because these methods rely on vehicles crossing a specific point, they often fail to register stopped cars, creating a 'zero flow' paradox where complete gridlock is incorrectly categorized as an empty, free-flowing road. To address these limitations, this study proposes an end-to-end computer vision framework utilizing the Real-Time DEtection TRansformer (RT-DETR) model to quantify vehicle volume and immediately classify a roadway's Level of Service (LOS) through spatial density. In processing video feeds from the Ateneo de Manila University Campus Safety and Mobility Office, the framework calculates an instantaneous, frame-level Spatial Density Index across six distinct vehicle classes. Results show that the proposed SDI framework outperforms the traditional time-windowed method in estimating traffic congestion and reduces percentage error margins to between -0.22% and 2.85%. The proposed framework successfully resolves the zero flow paradox by analyzing frame-wide congestion rather than fixed line crossings prescribed in the time-windowed method. It also offers a highly responsive and robust alternative that detects immediate spikes in vehicle density, which makes it potentially suitable for integration into intelligent traffic systems.**

***Keywords*—Traffic congestion estimation; level of service; volume-to-capacity ratio; RT-DETR; real-time object detection**

I. Introduction

Traffic congestion remains a critical challenge in many urban areas around the world. It affects the efficiency of transportation, quality of life, and economic productivity [1]. Due to growing urbanization, there is a need for intelligent transport systems that are capable of monitoring and analyzing traffic conditions in real-time [2]. Computer vision (CV), a field of artificial intelligence that is focused on enabling machines to extract meaningful information from visual data, has emerged as an effective solution to address these challenges [3].

Traffic congestion estimation plays a key role in managing and optimizing transport systems. It often involves analyzing vehicle volume and capacity to infer congestion levels [4]. Traditional methods for traffic congestion estimation utilize various technologies for collecting traffic information, such as inductive loop detectors, Light Detection and Ranging (LiDAR) sensors, and Global Positioning System (GPS)-based tracking [5], [6], [7]. However, these methods have drawbacks. GPS-based systems have limited coverage as they rely on vehicles being equipped with tracking devices [6]. Loop detectors require road installation and maintenance, which can be costly and disruptive [8]. LiDAR sensors' performance can be heavily affected by different weather conditions as they interfere with the laser signals and reduce accuracy [9].

With the increase in deployment of surveillance cameras across urban centers, their integration into intelligent transport system provides a highly beneficial method to address traffic congestion in real-world scenarios. Deep learning models in computer vision have shown significant promise in addressing challenges associated with traffic congestion [10]. Among its many applications, object detection, a technique for identifying and locating objects within images, has shown potential in enhancing current intelligent transport systems and providing a non-intrusive mechanism to quantify traffic flow.

Despite these capabilities, current CV approaches often fall short of providing a complete end-to-end analysis for traffic management. Existing frameworks typically output raw vehicle counts which require further manual processing to derive standard congestion measures [11]. Currently, congestion detection is treated as a visual classification problem through the discretization of the degrees of congestion [12]. Furthermore, few studies have shown a framework that would transform vehicle detection counts, obtained via CV-based solutions, into a measure of traffic congestion.

This study addresses this limitation by proposing an end-to-end framework that uses object detection to quantify vehicle volume and classify the Level of Service (LOS) of a roadway. While baseline evaluations utilize traditional time-windowed V/C ratios, this study introduces a Spatial Density Index (SDI) to address their inherent temporal lag. The proposed SDI provides an immediate classification of LOS through frame-level evaluation of spatial occupancy of a road by vehicles.

II. Review of Related Literature and Studies

A. Traffic Congestion Estimation

Traffic congestion is defined as a condition where the demand for road space exceeds the available capacity; this leads to a decrease in vehicle speed, longer travel times, and the overall inefficiency of the road network [8]. It can be caused by various factors such as high traffic volumes, accidents, roadworks, or inefficient traffic management. When congestion occurs, the flow of vehicles is hindered, causing

delays and contributing to environmental pollution through increased emissions [13].

To accurately estimate congestion, various parameters are considered. These parameters, such as traffic flow, speed, density, and travel time, are essential for assessing congestion, since they offer measurable data points that reflect real-time and overall road network performance [14].

For the purpose of this study, the parameters (see Table I) defined by the Philippine Department of Public Works and Highways (DPWH) Highway Planning Manual [4] for the estimation of traffic congestion were used to classify the level of congestion detected in traffic scenes. Level of Service is a qualitative measure describing the operational conditions within a traffic stream and their perception by motorists and passengers. It is categorized into different levels according to the value of the Volume-to-Capacity (V/C) ratio of the road, visualized in Fig. 1.

As for the quantitative parameters, Volume is defined as the number of vehicles passing at a point or section of roadway per lane for a given time period (number of vehicles/time), usually measured by hour. Volume is a weighted measure describing how much space those vehicles occupy, and that depends on the Flat Terrain Passenger Car Equivalent Factor (PCEF) (see Table II) of the different vehicle classes that compose the traffic. On the other hand, capacity is the maximum number of vehicles that have a reasonable expectation of passing over a given section of lane or roadway in one direction or in both directions during one hour under prevailing road and traffic conditions (see Table III). It is typically expressed as passenger car unit per hour (PCU/hr).



Fig. 1. LOS Examples [15].

TABLE I. LEVEL OF SERVICE FOR PHILIPPINE HIGHWAYS [4]

Level of Service	V/C Ratio	Description
A	< 0.20	Free Flow Traffic
B	0.21–0.50	Free Flow Traffic
C	0.51–0.70	Moderate Traffic
D	0.71–0.85	Moderate/Heavy Traffic
E	0.86–1.00	Heavy Traffic
F	> 1.00	Forced Flow, Stop and Go

The V/C ratio defined by the DPWH provides a standardized metric for congestion estimation. However, its reliance on fixed time-windows introduces a temporal lag. Since the V/C ratio is calculated using aggregate per-hour or per-minute data, the resulting LOS reflects the average past traffic states rather than real-time road conditions.

B. Vehicle Detection in Traffic Scenes

Vehicle detection is the process of identifying and recognizing vehicles within a specific environment, such as roadways or parking areas [16]. Accurate vehicle detection is important for estimating congestion because it provides real-time data on the number, position, and movement of vehicles on roadways. This information helps in calculating important traffic parameters for determining the degree of congestion [17]. Accurate vehicle detection allows traffic management systems to improve urban mobility by detecting bottlenecks and anticipating traffic jams, and facilitating the implementation of more effective intelligent transport systems such as adaptive traffic signal control or route recommendations to reduce congestion [18].

Traditional vehicle detection methods rely on sensor-based technologies that are installed within road infrastructure. These methods include inductive loop detectors (ILDs), infrared sensors, microwave radar, and LiDAR-based approaches, all of which have been widely used in intelligent transport systems for vehicle identification and tracking. Despite their effectiveness for basic tracking, these sensor-based methods face several challenges and limitations. ILDs and infrared sensors are effective for basic tracking, but they are hindered by high installation costs, road surface disruption, and sensitivity to environmental factors like fog or rain. Similarly, while microwave radar and LiDAR offer resilience in adverse weather and high-precision 3D mapping, they face challenges with stationary vehicle detection and significant computational and financial overhead. Additionally, many of these methods require physical infrastructure changes, which makes them difficult to deploy in rapidly evolving traffic environments.

To address these limitations, CV-based vehicle detection has emerged as a more flexible and scalable alternative. Advancements in image processing and machine learning have led to the adoption of CV techniques for vehicle detection. Vehicle detection in CV involves identifying and locating vehicles within images or video frames, typically captured in traffic scenes [19]. These methods primarily utilize camera feeds from surveillance systems. This addresses the challenge sensor-based methods face, as it eliminates the need for extensive physical infrastructure. Modern object detection algorithms, such as convolutional neural networks, have significantly improved the accuracy and efficiency of vehicle detection, making CV an increasingly viable solution for traffic monitoring, congestion estimation, and even autonomous vehicle systems [20].

However, despite the significant improvement from tradi-

TABLE II. PCEF VALUES [4]

Vehicle Type	Description	Flat Terrain PCEF
1	Motor tricycle	2.5
2	Passenger car, Motorcycle	1.0
3–5	Passenger and goods utility and small bus	1.5
6	Large bus	2.0
7	Rigid truck, 2 axles	2.0
8	Rigid truck, 3 axles	2.5
9	Truck semi-trailer, 3 and 4 axles	2.5
10	Truck semi-trailer, 5+ axles	2.5
11	Truck trailers, 4 axles	2.5
12	Truck trailers, 5+ axles	2.5

TABLE III. BASIC HOURLY CAPACITY, PCU [4]

Carriage Width (m)	Urban Hourly PCU
Single less than 4	600
4.0–5.0	1,200
5.1–6.0	1,600
6.1–6.7	1,700
6.8–7.3	1,800
2 × 6.7 or 2 × 7.3	6,700

tional methods, vision-based vehicle detection systems still face significant challenges that hinder its effectiveness in real-world traffic scenarios. One of the most challenging problems is occlusion, which occurs when portions of a vehicle are occluded by other objects like other cars, pedestrians, road construction, or natural environments like trees and shadows. The accuracy and reliability of vehicle detection systems can be heavily affected by occlusion, which can result in partial or complete vehicle misdetection, especially in dense traffic settings [21]. Additionally, viewpoint variations also pose a problem, as vehicles appear differently depending on the distance and angles of the cameras. While deep learning models have improved object detection significantly, they still struggle to generalize well under these conditions without extensive training on diverse datasets [22].

C. End-to-End DEtection TRansformers

The current CV landscape is dominated by several state-of-the-art architectures, most notably the YOLO (You Only Look Once) family, SAM (Segment Anything Model), and DETR (DEtection TRansformer) [23]. DETR-based models, with their global attention mechanisms, are inherently better equipped to handle the occlusions of vehicles in dense traffic scenes.

The Real-Time DEtection TRansfomer (RT-DETR) is a new approach to object detection that aims to address the limitations of current real-time detectors like the YOLO series and earlier DETRs. The architecture of RT-DETR consists of a backbone, a hybrid encoder, and a Transformer decoder with auxiliary prediction heads [24]. RT-DETR demonstrates state-of-the-art real-time object detection performance, outperforming previously advanced YOLO detectors in both speed (in terms of frames per second) and accuracy (in terms of mean average precision) on the Common Objects in Context (COCO) dataset [24]. Real-time, in this case, is defined as an object detector with an inference speed of 20 frames per second or 50 ms per frame [25]. RT-DETR also significantly outperforms other DETR variants in terms of speed, achieving comparable or better accuracy with higher frames per second [24].

III. METHODOLOGY

A. Image Dataset Creation

The experimental workflow began with the acquisition and preparation of the raw data from the Ateneo de Manila University Campus Safety and Mobility Office (CSMO) video feeds which were turned into a dataset.

For this study, the CSMO provided static video feeds from Gates 2, 2.9, 3, and 3.5 of the university, running at 15 frames per second (see Fig. 2). The feeds were taken on 2025-02-20, 06:00 AM to 2025-02-21, 06:00 AM; 2025-04-03, 06:00 AM to 2025-04-04, 06:00 AM; and 2025-10-07, 06:00 AM to 2025-10-08, 06:00 AM.

All of the frames were sampled using FFmpeg. To avoid excessively long training times while still ensuring sufficient representation of the video footage, two frame samples were taken per minute on the footage. For the training process, frames were only extracted from the feeds on 2025-02-20 to 2025-02-21 and 2025-04-03 to 2025-04-04. The footage on Gates 2.9, 3, and 3.5, frame samples were taken from 06:00 AM of the first date up until 10:00 PM, and from 05:00 AM of the succeeding date up until 06:00 AM. For the footage on Gate 2, the frame samples were taken from 06:00 AM on the first date up until 06:00 PM and from 05:00 AM on the succeeding date up until 06:00 AM. Conceptually, 11,760 frames should be sampled. However, due to the fragmented nature of the footage, a total of 11,783 frames were sampled by FFmpeg, with 23 extra samples. Since these 23 extra samples were only a very small proportion of the projected total of 11,760 frames, it was decided that the frames would still be kept for the annotation process.

For the testing process, frames were sampled from the footage on 2025-10-07 to 2025-10-08. For the footage on Gates 2.9, 3, and 3.5, samples were taken from 06:00 AM to 01:00 PM and 04:00 PM to 10:00 PM on 2025-10-07. For the footage on Gate 2, samples were taken from 06:00 AM to 01:00 PM and 02:00 PM to 06:00 PM on 2025-10-07, and 05:00 AM to 06:00 AM on 2025-10-08. Following this process, an additional 3,060 frames were sampled from the given footage with no extra frames. Combined together, there were a total of 14,843 sampled images in the dataset.

The sampled images were in 1920 x 1080 pixels, but was resized to 640 x 640 pixels when fed into the model for training and validation, similar to the image sizes in COCO dataset, which the base RT-DETR was trained on [24]. Moreover, having a consistent size for all images allows for more effective feature extraction as it helps effectively capture relevant details and patterns, which is especially important for detecting occlusions where details may be subtle.

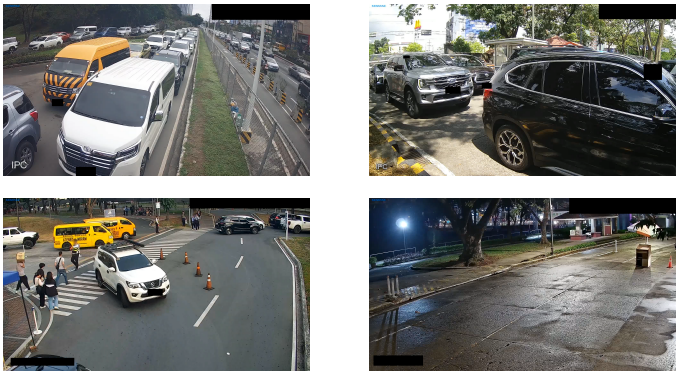


Fig. 2. Sample images from CSMO-provided feeds.

The sampled images were then labeled by an object annotation tool. Among the commonly used tools like CVAT, Roboflow Annotate, LabelStudio, and Make Sense, CVAT was

selected as it offers a free, open-source, and feature-packed solution for annotations [26].

In annotating the images, the six vehicle classes defined by the DPWH Highway Planning Manual [4], namely car, motorcycle, tricycle, bus, van, and truck were used. To guide the annotators in annotating the images, the sample images for each of the categories shown in the Philippines' Land Transportation Office [27] Filipino Driver's Manual were used as the visual basis.

The generated image annotations were saved in a JavaScript Object Notation (JSON) file format using the COCO annotation format. Sample annotation images can be seen in Fig. 3a and Fig. 3b.



(a) Congested traffic. (b) Freeflowing.

Fig. 3. Sample annotated frames.

B. Model Training and Testing

The RT-DETR model with the ResNet50 backbone was then trained on the training data subset with 11,760 frames and then tested on the testing data subset with 3,060 frames.

To test the counting and traffic congestion estimation capabilities of the model, video footage was taken and trimmed from the raw CSMO footage on 2025-10-07 to 2025-10-08. 12-hour segments from 06:00 AM to 06:00 PM were chosen from the video footage as these represent the operational hours of the school roads. After trimming these segments from the raw footage, the number of vehicles that passed through the gate were counted according to their category. These counts served as the ground truth versus the model's counting capability.

C. Vehicle Counting and Ground Truth Annotation

1) Ground truth validation: Manual vehicle counts were established by the researchers to serve as the ground truth. To ensure measurement reliability and mitigate annotation bias, the manual counts underwent a rigorous verification stage. Annotations were all performed by the researchers. Initial counts were self-validated two times by a primary annotator and subsequently cross-validated two times by a secondary annotator. In instances of discrepancy, the cross-validation count was established as the final ground truth.

2) Automated counting pipeline: The valid vehicle types to be counted were selected based on the availability of PCEF values in the DPWH Highway Planning Manual [4] and its specified classes, namely, cars, motorcycles, tricycles, buses, vans, and trucks.

To accommodate the comparative analysis, two distinct vehicle counting pipelines were implemented:

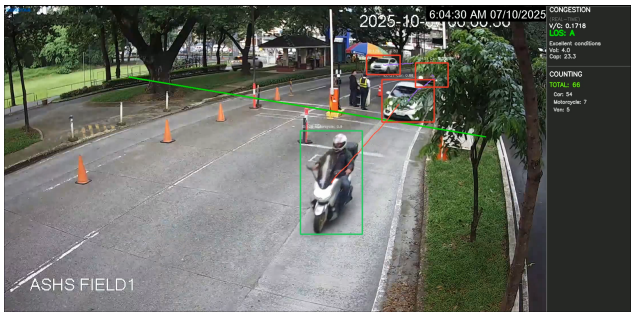


Fig. 4. Vehicle counting sample.

- Time-Windowed Pipeline: For the traditional metric, the ByteTrack Multi-Object Tracking Algorithm [6] was used. A vehicles was only counted when its tracked trajectory crossed a user-defined line on the roadway, as illustrated in Fig. 4.
- Frame-Level Spatial Pipeline: For the proposed spatial metric, the line-crossing logic was discarded. Instead, the model executed a frame-level count by aggregating all visible vehicles in a single frame simultaneously.

D. Traffic Estimation Metrics

In this study, two model frameworks were used to translate the automated vehicle detections and counts into congestion measures. This time-windowed framework was based on the standard Volume-to-Capacity ratio established by the DPWH. The LOS was determined based on equivalent traffic levels of the computed V/C ratio (see Table I).

1) Time-windowed volume-to-capacity ratio: The Volume-to-Capacity ratio is a quantitative measure of the demand of vehicles placed in a roadway compared to its ability to handle the traffic [4].

Under this time-windowed framework, raw vehicle counts are transformed into Passenger Car Units using the equivalent Passenger Car Equivalent Factors to account for the varying impact of different vehicle types on traffic flow [4]. Volume pertains to the number of vehicles that are using the road per hour. Volume is calculated by aggregating the weighted counts of six distinct vehicle classes: cars, vans, motorcycles, tricycles, buses, and trucks. Hence, the window-based volume is defined in Eq. 1:

$$\text{Volume}_{tw} = \sum_{i=1}^{6} (n_i \times PCEF_i) \tag{1}$$

where, n_i represents the raw count of vehicles in class i, and $PCEF_i$ represents the corresponding factor for that specific class. Note that in this implementation, volume is calculated over the past n-minutes.

On the other hand, capacity is the maximum number of vehicles per hour that the road can handle before congestion occurs. A roadway's capacity (Capacity_{tw}) is based on its carriage width. The DPWH [4] defines the hourly capacity for a given road's carriage width in Table III. Hence, the V/C ratio is defined in Eq. 2:

$$\text{V/C Ratio} = \frac{\text{Volume}_{tw}}{\text{Capacity}_{tw}} \tag{2}$$

The capacity was obtained via a user-defined road width which was mapped to its corresponding PCU. This capacity was then scaled down for the shorter time intervals used in the video inference.

2) Frame-level spatial density index: While the DPWH-based framework provides a standardized metric for assessing traffic, its reliance on fixed time-windows introduces a temporal lag. Because the V/C ratio is calculated using aggregate data from the preceding n-minutes or hours, the resulting LOS reflects past traffic states rather than current conditions.

To address the limitations of the time-windowed V/C ratio, a Spatial Density Index (SDI) was formulated based on the instantaneous frame-level spatial occupancy. In this approach, frame-based volume is defined in Eq. 3 as the weighted sum of vehicle counts detected within a single frame:

$$\text{Volume}_{fl} = \sum_{i=1}^{6} (n_{i,frame} \times PCEF_i) \tag{3}$$

Meanwhile, the modified capacity is defined in Eq. 4:

$$\text{Capacity}_{fl} = k_j \times n_l \times L \tag{4}$$

where, k_j is the jam density, n_l is the number of lanes in focus, L is the length of the roadway in focus in kilometers. Jam density is defined as the maximum possible vehicle density on the road when traffic is completely stopped [28]; its units are expressed in veh/km/lane. Theoretical simulations by Knoop and Daamen [28] identified that the theoretical jam density was 145 veh/km/lane.

The resulting SDI is then expressed in Eq. 5:

$$\text{SDI} = \frac{\text{Volume}_{fl}}{\text{Capacity}_{fl}} \tag{5}$$

Both of these ratios are dimensionless since both the PCE-weighted numerator and capacity denominator share the same unit, namely the number of vehicles.

3) Percent error: Accuracy is an important evaluation metric that can be used to evaluate the performance of the proposed method. Percent error represents the difference between an estimated value and the true value as a percentage of the true value [29]. It can be signed when analyzing bias or unsigned when expressing accuracy. It can be computed using the formula,

$$\text{Signed \% Error} = \frac{\text{Estimated Value} - \text{True Value}}{\text{True Value}} \times 100\% \tag{6}$$

For this study, this metric will be used to provide insights into the model's ability to underestimate or overestimate in its counting of vehicles and computation of the level of traffic congestion estimation based on the Volume-to-Capacity ratio or the Spatial Density Index.

IV. Results and Discussion

A. Base Model Performance and Inference Latency

Before evaluating the congestion estimation pipelines, the underlying RT-DETR model's object detection reliability was briefly assessed. As seen in Table IV, the model demonstrated highly robust baseline detection capabilities for the selected vehicle classes.

TABLE IV. Model Performance of RT-DETR

Precision	Recall	mAP@0.5	mAP@0.5:0.95
0.999	0.989	0.934	0.795

To validate the real-time applicability of both the traditional and proposed end-to-end framework, the computational complexity of the model was measured. Using an NVIDIA RTX 3090 GPU for the complete pipeline, it can be noted that the system achieved an inference speed of 29.59 frames per second (FPS) (see Table V). This exceeds the standard 20 FPS threshold required for real-time traffic monitoring systems [25].

TABLE V. Computational Complexity of RT-DETR

Parameters	GFLOPs	Inference (FPS)	Memory (GB)
42,710,790	67.91	29.59	2.121

B. Time-Windowed Traffic Congestion Estimation

The model's ability to count vehicles over 12 hours (06:00 AM to 06:00 PM) was evaluated against the manually counted ground truth, as shown in Table VI. Across most gates, the model showed a strong performance in counting the number of vehicles, maintaining an error rate of around $\pm 10\%$. However, Gate 2 showed a significant outlier with a -15.84% error in vehicle counting.

TABLE VI. Comparison of the Ground Truth, and RT-DETR Vehicle Count

Gate	Ground Truth	RT-DETR	% Error
Gate 2	3953	3643	-7.84%
Gate 2.9	5171	5303	2.55%
Gate 3	7442	7396	-0.62%
Gate 3.2	2424	2556	5.45%
Gate 3.5	4084	4261	4.33%

TABLE VII. Comparison of the Ground Truth, and RT-DETR V/C Ratio (Time-Windowed)

Gate	Ground Truth	RT-DETR	% Error
Gate 2	0.196	0.157	-19.97%
Gate 2.9	0.448	0.453	1.12%
Gate 3	0.304	0.294	-3.23%
Gate 3.2	0.145	0.135	-6.75%
Gate 3.5	0.978	0.973	-0.51%

The V/C ratios derived from the counts in Table VII typically had error margins within -6.75% to 1.12%. Gate 2

was also a notable exception with a -19.97% error in the V/C ratio.

C. Strengths and Limitations of the Time-Windowed Method

The traditional time-windowed method [4] shows significant strengths primarily oriented toward long-term planning as the V/C ratio, as defined in this method, can be used to predict the exact years that a roadway may reach a saturation point and require engineering interventions [30]. The primary strength of this approach is its focus on average throughput over a long period, such as an hour. As such, it is able to ignore brief fluctuations such as drivers stopping. Hence, the primary utility of the method is in the analysis of historical congestion data to identify persistent trends.

However, the method has significant limitations for real-time applications. One critical flaw is its reliance on point-based volume counts. When the road is empty, only a few vehicles or no vehicles will pass over the counting line. Similarly, when the road is completely gridlocked, only a few vehicles or no vehicles will also pass over the counting line. In both cases, the volume and, consequently, the V/C ratio become 0, which is equivalent to LOS A. Therefore, an empty and a gridlocked roadway are both considered as free-flowing traffic, resulting in the 'zero flow' paradox (see Fig. 7). The paradox lies in the inability of the method to distinguish between the two scenarios without supporting context from, for example, a live camera. Furthermore, the traditional V/C ratio requires calculation over n-minute or hour intervals to obtain an average flow rate. This delay means V/C ratios fail to reflect rapid changes in congestion, making them unsuitable for real-time control. As seen in Fig. 5, while the average vehicle count from 3:00 PM to 3:10 PM is at a high 16.59, the average LOS was calculated at A, indicating free flow. This means that at 3:00 PM to 3:10 PM, it was not able to properly represent the LOS of the past 10 minutes as no cars were moving past the counting line.

A notable outlier observed in both the vehicle counts and the V/C ratio computations was Gate 2, where the error was significantly higher. Upon checking the video feed, the camera at Gate 2 was observed to be positioned in a ground-level, side-view manner, with vehicles in the foreground frequently occluding those in the background. This left the model with insufficient visual features to identify and count trailing vehicles. This highlights the critical impact of camera placement on model detection accuracy, as accuracy relies on an object having the appropriate resolution at a given distance [31]. Going beyond the optimal distance makes objects look smaller and more difficult to separate from noise [31]. Moreover, optimal placement ensures that the maximum possible area is covered while minimizing blind spots or redundant overlaps between cameras [32]. Hence, it can be observed here how improper placement led to significant undercounting, which skewed the Level of Service computations for a given gate.

D. Frame-Level Traffic Congestion Estimation

The proposed frame-level framework demonstrates a notable performance improvement over the traditional time-windowed approach. As shown in Table VIII, the proposed

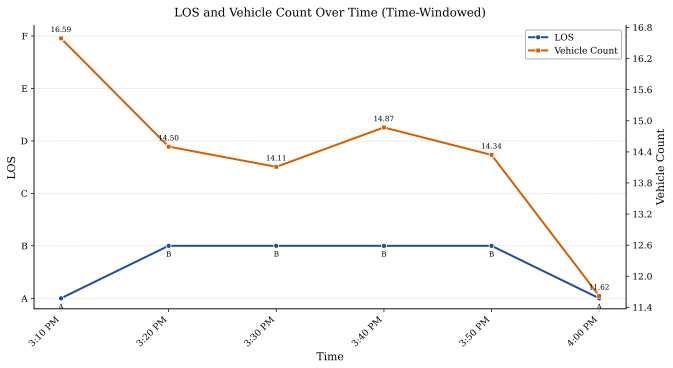


Fig. 5. LOS and vehicle count over time (time-windowed).

TABLE VIII. COMPARISON OF THE GROUND TRUTH AND RT-DETR SDI (FRAME-LEVEL)

Gate	Ground Truth	RT-DETR	% Error
Gate 2	0.400	0.355	-11.08%
Gate 2.9	0.794	0.815	2.69%
Gate 3	0.179	0.178	-0.77%
Gate 3.5	0.334	0.333	-0.22%

method reduced the percentage error margin to between -0.22% and 2.85%, a substantial improvement over the time-windowed measurements with range of -6.75% to 1.12%. Notably, the framework also demonstrated resilience to suboptimal camera configurations in its LOS computations compared to the time-windowed method. For instance, at Gate 2, despite a highly problematic viewing angle, the model reduced the error relative to the ground truth from a baseline of -19.97% in the time-windowed method to -11.08%. While the negative percentage value indicates that there is still an underestimation in congestion, the proposed framework mitigates the severe miscalculations inherent in the time-windowed method.

E. Strengths and Limitations of the Frame-Level Estimation Method

A primary strength of the proposed method is its ability to solve the zero flow paradox. This proposed method, by analyzing the entire frame, instantly distinguishes between gridlocked and free-flowing traffic states by detecting the presence of stationary vehicles (see Fig. 7). As seen in Fig. 6, the average vehicle count at the interval from 3:00 to 3:10 PM is 16.59, with the LOS being at level D. When contrasted with Fig. 5, this shows that the frame-level method can distinguish a congested road, whereas the time-windowed method cannot. Furthermore, the method provides a significant advantage in temporal resolution versus the time-windowed method. The proposed method is able to detect spikes in vehicle density immediately, offering data that is far more responsive for applications like adaptive traffic signal control.

However, the method has limitations dependent on camera perspective and scene complexity. Because it detects all vehicles within the entire video frame, it can be inaccurate for frames containing multiple traffic routes, such as in Gates 3.2 and 3.5, where distinguishing the specific lane or road segment is challenging. Without Region of Interest (ROI) masking, the

model may aggregate the traffic density of the entire frame, making it difficult to isolate and assign specific congestion metrics to each individual gate. Another significant constraint is occlusion. A common issue with single, oblique-angle CCTV views is that larger vehicles, like trucks, can obscure smaller vehicles behind them, such as cars or motorcycles. This leads to an underestimation of true density. A top-down perspective from a drone or satellite would mitigate this [33], but it is often not available for most traffic monitoring infrastructures.

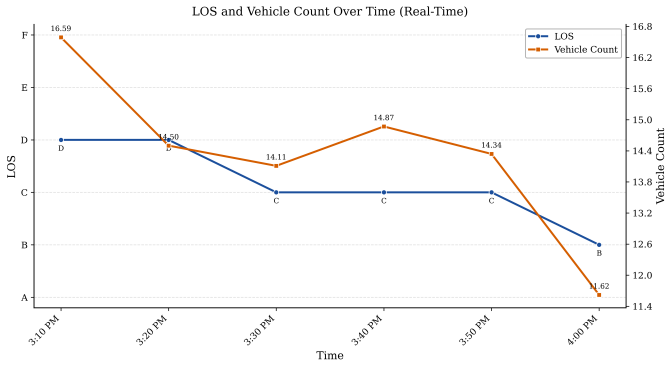


Fig. 6. LOS and vehicle count over time (frame-level).



Fig. 7. Time-windowed method (top) vs. frame-level method (bottom) detection comparison in congested states.

V. Conclusion

In conclusion, this study demonstrates that frame-wide spatial density analysis offers a robust alternative to the traditional time-windowed and line-based traffic congestion estimation by directly resolving the zero flow paradox inherent in severely congested conditions. By processing instantaneous vehicle density through the proposed frame-level method, the framework outperformed the traditional methods and effectively reduced percentage errors by approximately half while maintaining a more accurate representation of the true congestion state. Furthermore, unlike traditional flow-based V/C ratios that require substantial time-windows, the ability of the SDI to detect immediate spikes in congestion makes it highly applicable for integration into adaptive traffic signal control systems as it uses real-time congestion data; this enables control systems to autonomously optimize signal timing and alleviate sudden congestion [34].

VI. Recommendations for Future Work

The current framework utilizes jam density thresholds derived from the literature based on traffic conditions in the Netherlands. However, Philippine traffic is highly heterogeneous, characterized by unique vehicle dimensions (e.g., jeepneys, motorcyles, etc.), unstructured lane discipline, and tighter vehicle packing during severe congestion. Consequently, applying foreign baselines may skew maximum density estimations. Future research should consider establishing localized jam density metrics. Defining these local parameters will critically calibrate the model's congestion thresholds and improve the accuracy of Level of Service (LOS) classifications in similar developing urban centers.

To address the limitations of processing multi-directional traffic within a single camera feed, future research should investigate integrating ROI masking. Applying explicit spatial boundaries will allow the model to isolate specific road segments and lane directions. This targeted extraction is crucial for preventing data aggregation in complex scenes and enabling the framework to provide accurate, road-specific congestion estimations rather than generalized frame-wide density.

To address the limitation of vehicle occlusion inherent in ground-level camera angles, future implementations may look into prioritizing top-down camera perspectives. As noted, ground angles frequently result in larger vehicles (e.g. vans, trucks) obscuring smaller trailing vehicles, which inherently leads to an underestimation of true traffic congestion.

While this study successfully serves as a proof-of-concept for the SDI, certain methodological limitations must be acknowledged. First, due to the limited sampling windows of the university vehicular gates, the current evaluation lacks the statistical volume necessary for cross-validation. Consequently, the reported error margins require further validation across diverse environmental conditions, including varying weather and lighting. Second, the SDI has to be adapted into discrete DPWH Level of Service categories since the LOS categories outlined by the DPWH are based on the V/C ratio. Future studies must consider both of these before considering the generalized deployment of this system.

Acknowledgment

The authors would like to thank the Campus Safety and Mobility Office (CSMO) for providing the dataset used in this study.

Special thanks are extended to Social and Environmental Engagement for Development and Sustainability (SEEDS) for the provision of the computer used for model training and to Research, Creative Work, and Innovation (RCWI) for their administrative assistance and organizational support throughout this research.

VII. Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini 3.1 Pro to improve the clarity of the paper and aid in proofreading. After the use of this tool, the authors carefully reviewed and edited the content as needed to align with the SAI paper format. The authors take full responsibility and accountability for the content in this work.

References

[1] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," *Journal of Advanced Transportation*, vol. 2021, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1155/2021/8878011

[2] R. Ravish and S. R. Swamy, "Intelligent traffic management: A review of challenges, solutions, and future perspectives," *Transport and Telecommunication Journal*, vol. 22, no. 2, pp. 163–182, Apr. 2021. [Online]. Available: https://doi.org/10.2478/ttj-2021-0013

[3] H. Ghahremannezhad, H. Shi, and C. Liu, "Object detection in traffic videos: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6780–6799, Jul. 2023. [Online]. Available: https://doi.org/10.1109/TITS.2023.3258683

[4] D. of Public Works and Highways, "Highway planning manual," 2007.

[5] R. R. Sai, A. K. Tangirala, and L. Vanajakshi, "Road traffic analysis using 2d lidar," in *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, ser. COMSNETS 2024. New York, NY, USA: IEEE, 2024, pp. 228–233. [Online]. Available: https://doi.org/10.1109/COMSNETS59351.2024.10426892

[6] Z. Zhang, S. Zhang, L. Mo, M. Guo, F. Liu, and X. Qi, "Traffic volume estimate based on low penetration connected vehicle data at signalized intersections: A bayesian deduction approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 596–10 609, Aug. 2022. [Online]. Available: https://doi.org/10.1109/TITS.2021.3094933

[7] N. K. Singh, A. K. Tangirala, and L. D. Vanajakshi, "A multivariate analysis framework for vehicle detection from loop data under heterogeneous and less lane disciplined traffic," *IEEE Access*, vol. 9, pp. 143 580–143 591, Oct. 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3120470

[8] F. H. A. Research and Technology, "Chapter 1, traffic detector handbook: Third edition—volume i - fhwa-hrt-06-108," 2006, retrieved from https://www.fhwa.dot.gov/publications/research/operations/its/06108/01.cfm. [Online]. Available: https://www.fhwa.dot.gov/publications/research/operations/its/06108/01.cfm

[9] J. Kim, B.-j. Park, and J. Kim, "Empirical analysis of autonomous vehicle's lidar detection performance degradation for actual road driving in rain and fog," *Sensors*, vol. 23, no. 6, Mar. 2023. [Online]. Available: https://doi.org/10.3390/s23062972

[10] Q. Xue, "Advancements in object detection: From machine learning to deep learning paradigms," in *Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (CONF-SEML 2024)*, ser. CONF-SEML 2024. Oxford, United Kingdom: EWA Publishing, 2024, pp. 154–159. [Online]. Available: https://doi.org/10.54254/2755-2721/75/20240530

[11] N. R. Dangal, D. Wangmo, S. Meena, and S. Kumar, "Study of traffic flow for estimation of level of service," *International Journal for Research Trends and Innovation*, vol. 8, no. 4, pp. 1348–1353, Apr. 2023. [Online]. Available: https://www.ijrti.org/viewpaperforall?paper=IJRTI2304220

[12] X. Li, T. Hao, X. Jin, B. Huang, and J. Liang, "Fine traffic congestion detection with hierarchical description," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 439–24 453, Dec. 2022. [Online]. Available: https://doi.org/10.1109/TITS.2022.3206583

[13] P. Krütli, D. Nef, P. Kangethe, D. Etongo, and M. Stauffacher, "Sustainable land use – seychelles' transport system," ETH Zurich, Zurich, Switzerland, Tech. Rep., 2022.

[14] M. Z. Mehdi, H. M. Kammoun, N. G. Benayed, D. Sellami, and A. D. Masmoudi, "Entropy-based traffic flow labeling for cnn-based traffic congestion prediction from meta-parameters," *IEEE Access*, vol. 10, pp. 16 123–16 133, Feb. 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3149059

[15] T. R. Board, "Us highway capacity manual," 2010. [Online]. Available: https://www.jpautoceste.ba/wp-content/uploads/2022/05/Highway-Capacity-Manual-2010-PDFDrive-.pdf

[16] Z. Yang and L. S. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image and Vision Computing*, vol. 69, pp. 143–154, Jan. 2018. [Online]. Available: https://doi.org/10.1016/j.imavis.2017.09.008

[17] R. Bauza and J. Gozalvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295–1307, Sep. 2013. [Online]. Available: thttps://doi.org/10.1016/j.jnca.2012.02.007

[18] L. E. Y. Mimbela and L. A. Klein, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems," 2007. [Online]. Available: https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/vdstits2007.pdf

[19] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review*, vol. 11, Dec. 2019. [Online]. Available: https://doi.org/10.1186/s12544-019-0390-4

[20] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, and Z. Feng, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, Sep. 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2939201

[21] S. Gilroy, E. Jones, and M. Glavin, "Overcoming occlusion in the automotive environment—a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 23–35, Jan. 2021. [Online]. Available: https://doi.org/10.1109/TITS.2019.2956813

[22] Z. Deng, H. Sun, S. Zhou, J. Zhao, and H. Zou, "Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 8, pp. 3652–3664, May 2017. [Online]. Available: https://doi.org/10.1109/JSTARS.2017.2694890

[23] X. M. Li, X. H. Pu, W. C. Ling, and X. D. Song, "Yolo-sam an end-to-end framework for efficient real time object detection and segmentation," *Scientfic Reports*, vol. 15, Nov. 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-24576-6

[24] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR 2024. New York, NY, USA: IEEE, 2024, pp. 16 965–16 974. [Online]. Available: https://doi.org/10.1109/CVPR52733.2024.01605

[25] V. K. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," *Computer Science Review*, vol. 38, Nov. 2020. [Online]. Available: https://doi.org/10.1016/j.cosrev.2020.100301

[26] CVAT.ai, "Computer vision annotation tool (cvat)," 2018, retrieved from https://github.com/cvat-ai/cvat. [Online]. Available: https://github.com/cvat-ai/cvat

[27] P. L. T. Office, "Filipino driver's manual," 2021. [Online]. Available: https://lto.gov.ph/wp-content/uploads/2023/10/FDM-Vol.-1-2nd-Edition.pdf

[28] V. L. Knoop and D. Winnie, "Automatic fitting procedure for the fundamental diagram," *Transportmetrica B: Transport Dynamics*, vol. 5, no. 2, pp. 129–144, Nov. 2016. [Online]. Available: https://doi.org/10.1080/21680566.2016.1256239

[29] LibreTextsChemistry, "3.13: Percent error." [Online]. Available: https://chem.libretexts.org/Bookshelves/Introductory_Chemistry/Introductory_Chemistry_(CK-12)/03%3A_Measurements/3.13%3A_Percent_Error

[30] J. Arliansyah, M. R. Prasetyo, and A. Y. Kurnia, "Planning of city transportation infrastructure based on macro simulation model," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 4, pp. 1262—-1267, Aug. 2017. [Online]. Available: https://doi.org/10.18517/ijaseit.7.4.2444

[31] AllGoVision, "Optimal camera placement for object detection accuracy," AllGoVision, Bangalore, India, Tech. Rep., n.

[32] D. Khanal, "Traffic monitoring camera locations optimization study for interstate in idot district 8," Master's thesis, Southern Illinois University Edwardsville, 2023.

[33] X. Liu and Z. Zhang, "A vision-based target detection, tracking, and positioning algorithm for unmanned aerial vehicle," *Wireless Communications and Mobile Computing*, vol. 2021, 2021. [Online]. Available: https://doi.org/10.1155/2021/5565589

[34] M. Gupta, H. Miglani, P. Deo, and A. Barhatte, "Real-time traffic control and monitoring," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 5, p. 100211, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772671123001067

Beyond the Login Screen: Teaching Full-Stack Defense-in-Depth from Software to Hardware Logic

Md Mufassal Ahmad[1], Mary Nusrat[2], Gahangir Hossain[3]
Department of Electrical Engineering, University of North Texas, Denton, Texas, USA[1]
Department of Information Science, University of North Texas, Denton, Texas, USA[2]
Department of Data Science, University of North Texas, Denton, Texas, USA[3]

***Abstract*—Defense-in-Depth (DiD) holds that a system should be protected by multiple independent layers of security, so that the failure or compromise of a single countermeasure does not result in total system compromise. Secondary cybersecurity instruction, however, often presents security as a list of disconnected rules. This study presents the design of an instructional framework that teaches DiD to high-school students as a transferable form of systems thinking. The framework is developed within a teacher professional-development project and comprises five stages: Hook, Layer Mapping, Threat Modeling, Hardware Case Study, and Design Challenge, each grounded in an established learning theory and tied to six measurable, Bloom-aligned learning objectives. Threat modeling is introduced through a four-question process supported by student-facing adaptation of the six STRIDE categories and simple attack trees. A deliberately compact case study of the six-layer defense of logic-locked integrated circuits serves as a far-transfer task rather than as a hardware tutorial. The framework includes five assessment instruments, a five-criterion, four-level analytic rubric (5–20 points), and an implementation guide that fits the module into seven to nine 45–50 minute class periods. The contribution is primarily design-oriented: specifying three hypotheses, associated metrics, and a pretest–posttest protocol for a planned classroom pilot. By connecting familiar security protections with hardware security, DiD shifts the student's question from "Is the system secure?" to "How does the system remain resilient when an individual protection fails?".**

***Keywords*—Defense-in-depth; cybersecurity education; secondary education; threat modeling; STRIDE; instructional design; hardware security; logic locking**

I. Introduction

High-school students use digital systems every day and are exposed to the corresponding risks. Weak or reused passwords compromise personal accounts, phishing messages steal credentials, unpatched devices spread malware, and insecure networks leak sensitive information. Schools themselves are targets: the learning management system used by many districts was recently breached twice within a short period [1]. Cybersecurity education therefore needs to move beyond teaching security as a set of discrete actions or tools. Students need to understand security as a system in which several measures combine to reduce risk [2], [3].

Defense-in-Depth (DiD) therefore provides an appropriate organizing concept for cybersecurity instruction. It assumes that no single countermeasure can effectively address all possible threats [4]. For example, a strong password may reduce the likelihood of successful guessing attacks, multi-factor authentication may limit the consequences of compromised credentials, and login alerts may help detect unauthorized access attempts. Each mechanism serves a distinct defensive function, and their combined use provides greater resilience than reliance on any individual safeguard alone. The same principle applies to more advanced security domains. Rahman *et al.* [5] demonstrate that protecting the secret key of a logic-locked integrated circuit (IC) requires multiple independent safeguards against threats such as reverse engineering, optical probing, electrical probing, scan-chain attacks, and oracle attacks. Although the underlying techniques are technically complex, the core principle remains straightforward: valuable assets require multiple independent layers of protection because adversaries may exploit different attack paths.

For secondary education, DiD offers three key advantages. First, it connects directly to students' everyday use of technology. Second, it promotes systems thinking by requiring students to identify assets, threats, vulnerabilities, countermeasures, and potential failure paths. Third, it bridges familiar cybersecurity practices with engineering-oriented security design, enabling teachers to present cybersecurity as a structured reasoning process rather than as a checklist of isolated rules.

A. Scope and Addressing Design Gaps

This work is an instructional-design study. Following the design-based research tradition, in which a theoretically grounded intervention is specified before it is evaluated in iterative classroom cycles [6]. The study documents the design phase and is organized around three design questions:

- **DQ1.** How can DiD be decomposed into a sequence of learning activities that is accessible to students with no prior security training?
- **DQ2.** How can an advanced hardware-security case be reduced so that it supports transfer of DiD reasoning instead of becoming a technical tutorial?
- **DQ3.** Which instruments and criteria make DiD reasoning-asset identification, threat identification, layer diversity, and failure-path analysis, observable and assessable by a classroom teacher?

The hypotheses that follow from this design, and the protocol for testing them, are stated in Section VIII.

B. Contributions

- Six measurable learning objectives aligned with Bloom taxonomy and mapped to stages and instruments (Table I, Fig. 4).
- A five-stage instructional sequence for DiD in which every stage is justified by an established learning theory (Table II, Fig. 3).
- A student-facing adaptation of STRIDE and attack trees that gives the threat-modeling stage methodological depth (Table III, Fig. 4).
- A compact logic-locking case study used as a far-transfer task (Table IV, Fig. 5).
- An assessment plan with an analytic rubric and performance criteria (Tables V and VI), and an implementation guide covering duration, prerequisites, preparation, and resources (Table VII).

The remainder of the study is organized as follows. Section II reviews related work and the learning theories that ground the design. Section III defines DiD and explains why it suits secondary learners. Section IV describes the instructional context and the learning objectives. Section V presents the five-stage framework. Section VI presents assessment, Section VII addresses classroom implementation, Section VIII states limitations and the evaluation plan, and Section IX concludes the work.

II. Related Work and Theoretical Foundations

A. Cybersecurity Education in K-12

A growing body of work argues for integrating cybersecurity into K–12 curricula. Pencheva *et al.* [3] argue that cybersecurity should be woven into secondary education to foster a security mindset, and note that many programs present security as a collection of rules. Kamaludeen *et al.* [7] propose a cyber-protection framework for the K–12 sector that spans technical, administrative, and behavioral defenses. Vajpayee and Hossain [8] examine authentic assessment in high-school cybersecurity and stress that students must learn to evaluate cyber assets and risks holistically. Recent systematic reviews confirm that the field is active but uneven. Ibrahim *et al.* [9] find that most aspects of cybersecurity are not taught systematically at K–12 level; Sağlam *et al.* [11] report that content varies widely between countries and leaves gap; Arishi *et al.* [12] identify games, mobile applications, and tailored programs as effective awareness practices; and Xu and Li [13] map trends, challenges, and opportunities for younger learners. Across these reviews the dominant topics are awareness and safe behavior. Layered-defense reasoning is rarely an explicit learning goal.

B. Teaching Threat Modeling

Threat modeling is well developed in professional practice. STRIDE classifies threats into six categories [15], attack trees decompose an attacker's goal into alternative paths [16], PASTA defines a seven-stage risk-centric process [17], and surveys compare these and other methods [18]. Educational use has been studied mainly at university level. Joshi *et al.* [19] found that software-engineering students who received systems-thinking instruction together with STRIDE identified system-level as well as component-level threats, whereas students taught STRIDE alone concentrated on components. That result supports the premise of this study, that threat categories should be taught inside a systems view, but no comparable adaptation exists for secondary learners.

C. Learning Theories Informing the Design

Five strands of learning theory inform the proposed framework. *Constructivism* holds that learners build new understanding on prior knowledge, and that learning is most productive within the zone of proximal development, where support from a teacher or peer is available [20]. *Inquiry-based learning* organizes instruction as a cycle of orientation, conceptualization, investigation, conclusion, and discussion [21]. *Cognitive apprenticeship* makes expert reasoning visible through modeling, coaching, scaffolding, and fading [22], [23]. *Experiential learning* describes a cycle of concrete experience, reflective observation, abstract conceptualization, and active experimentation [24]. Research on *transfer learning* shows that learners abstract a general schema more reliably when they compare two analogous cases than when they study one [25], [26]. Section V assigns each stage to the theory that motivates it, and the revised Bloom taxonomy [27], [28] is used to state objectives and to order the stages by cognitive demand [29].

D. The Gap

Existing K–12 research establishes the importance of layered-defense thinking; however, it does not provide a scaffolded, theory-grounded instructional sequence with clearly assessable outcomes. While existing threat-modeling pedagogy primarily targets university students, this work addresses both gaps and incorporates a hardware-security case to examine whether DiD reasoning can transfer to an unfamiliar system.

III. Defense-in-Depth for Secondary Learners

A. Principle

DiD deploys multiple independent layers of security to reduce the risk that the failure of one component compromises the entire system. The concept originates from military strategy, in which successive defensive lines delay, weaken, and contain an adversary, and has since become a widely adopted principle in information and industrial control system security [4], [30], [31]. Each layer is designed to prevent, detect, contain, or facilitate recovery from specific categories of attack. A strong password resists guessing but cannot stop phishing or malware. Similarly, a firewall can block many network-based attacks but cannot prevent a user from entering credentials into a fraudulent website.

An essential distinction between *redundancy* and *diversity*, is that redundant layers repeat a similar protection, whereas diverse layers address different attack vectors [4], [5]. Students often assume that a strong password means a stronger system. A system that has only password system, with no added recovery measures, no device security, and no user awareness, remains fragile however strong the password is. Fig. 1 contrasts the single vs multi-layered defense configurations.

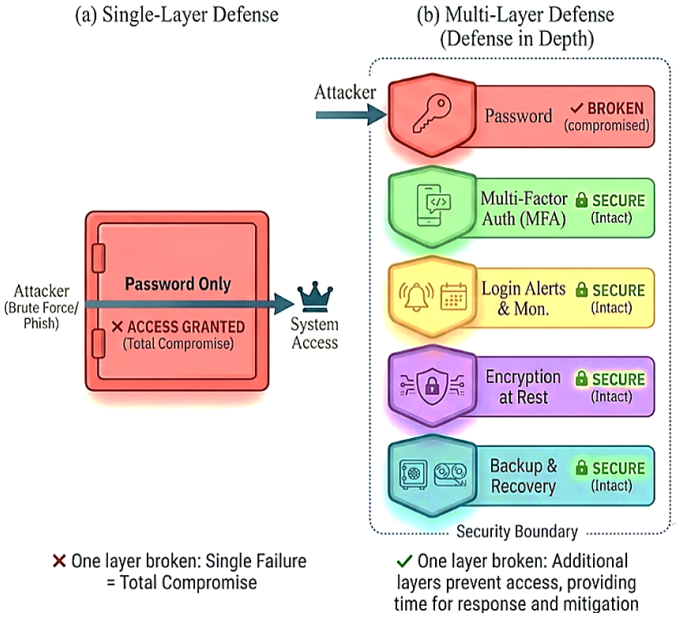


Fig. 1. Single-layer defense versus multi-layered defense.

B. Rationale for the Secondary Level

Secondary-level students increasingly rely on information systems for communication, learning, financial activities, social interactions and entertainment [7], [8], however, cybersecurity is often encountered as a set of disconnected rules rather than as an integrated component of their digital activities [10]. The DiD instructional approach targets three primary outcomes. The first outcome is *awareness*: students who understand that successful cyberattacks often exploit a single unprotected point may be more motivated to adopt multiple protective measures. The second is *systems thinking*: DiD requires students to identify what must be protected, how it could be compromised, which safeguards are already in place, and where gaps remain. The third is *transfer*: the layered reasoning developed through familiar contexts, such as email accounts, can be extended to more complex environments, including enterprise systems, critical infrastructure, and hardware. DiD is well suited to this age group because it does not require advanced technical skills [2] and can be taught through scenarios that are familiar to students.

C. Everyday Examples used in Instruction

Three familiar systems supply the examples used in Stages I to III.

1) Personal accounts: A student who reuses one password relies on a single layer. A layered alternative combines a unique password held in a password manager, multi-factor authentication (MFA), login alerts, securely stored recovery codes, and phishing awareness. Each layer answers a different threat: guessing, stolen credentials, undetected intrusion, loss of access, and deception.

2) School networks and devices: Schools combine account controls, web filtering, antivirus software, patching, network segmentation, backups, and monitoring [7], [31]. Patching

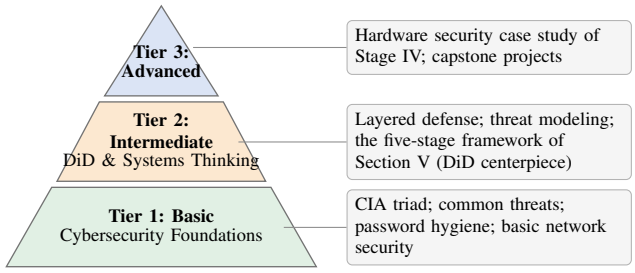


Fig. 2. Three-tier cybersecurity training for STEM teachers in the PERFECT project; the DiD framework forms Tier 2 and its capstone case belongs to Tier 3.

closes vulnerabilities, antivirus detects malicious files, segmentation contains an infection, backups enable recovery from ransomware, and monitoring reveals unusual activity. The example shows that security is organizational as well as individual, and that prevention, detection, containment, and recovery are different functions.

3) Human behavior: Students often perceive that cybersecurity is purely technical domain [3]. However, cyberattacks frequently exploit human factors such as trust, fear, curiosity, or ignorance, and a technically sound system fails when users unintentionally bypass its protections [14]. Accordingly, the module emphasizes the user as an integral layer of defense rather than as an entity external to the security system.

IV. Instructional Context and Learning Objectives

A. The PERFECT Project

The framework was developed within PERFECT (Providing Environment and Resources for STEM 9th–12th Graders in Effective Cybersecurity Training), a project that prepares high-school STEM teachers to teach cybersecurity [32], [33]. The project follows a train-the-trainer model with three tiers, shown in Fig. 2. Tier 1 covers foundational concept such as the confidentiality-integrity-availability triad, common cyberthreats, and password hygiene. Tier 2 covers DiD and systems thinking, and the five-stage framework presented in this work. Tier 3 covers advanced case studies, including the hardware case in Stage IV. The framework therefore has a dual use: it is a teacher-training module and, with minor adaptation, a lesson sequence for the high-school classroom.

B. Learning Objectives

Table I presents the six learning objectives (LOs), begins with an observable verb drawn from the revised Bloom's taxonomy [28] and is assessed through at least one instrument described in Section VI. The objectives are aligned with the Grade 9-12 cybersecurity expectations of the CSTA standards, which emphasize recommending and comparing appropriate security measures for specific scenarios [34], [35], as well as with the NICE's framework emphasis on task-oriented cybersecurity competencies [36].

TABLE I. MEASURABLE LEARNING OBJECTIVES

LO	On completing the module, students can...	Bloom level	Stage
1	Explain DiD and why a single layer is a single point of failure	Understand	I
2	Classify the protections of a familiar system by the threat addressed and by function (prevent, detect, contain, recover)	Analyze	II
3	Construct a threat model using the four questions, the STRIDE prompts, and an attack tree	Apply, Analyze	III
4	Match the six IC defense layers to the attacks they stop and to everyday analogues	Analyze (transfer)	IV
5	Design a layered defense plan for a system of their choice	Create	V
6	Evaluate a defense plan for diversity versus redundancy and trace its failure paths	Evaluate	III, V

V. THE FIVE-STAGE INSTRUCTIONAL FRAMEWORK

The framework moves students from intuition, through structured analysis, to independent design. Fig. 3 shows the five stage progression, and Table II summarizes, for every stage, the guiding question, the DiD principle, the student output, and the learning theory that motivates the stage. Cognitive demand rises from *understand* to *create* [28], and teacher support is withdrawn gradually, as cognitive apprenticeship prescribed in stages [22].

A. Stage I: Hook and Motivation

Stage I introduces DiD through relatable failures, for example an account taken over through a weak password, a school computer infected through an email attachment, or a ransomware incident that began on one unpatched machine [8]. No security vocabulary is required. The instructional goal is to make the fragility of single-layer protection visible, so that students leave with a question of their own: many failures occur not because every protection failed, but because the system depended on one. Stage II then turns from failures to the protections that students already have.

B. Stage II: Layer Mapping

Students identify and document the security measures implemented across systems they regularly use, including smartphones, school accounts, laptops, cloud storage services, and home Wi-Fi networks [7], [37]. Common examples include passwords, biometric authentication, multi-factor authentication (MFA), software updates, antivirus software, login notifications, data backups, and permission settings. Students then classify each security measure according to two dimensions: the threat it is intended to address and its defensive function-prevention, detection, containment, or recovery. This dual classification helps learners to recognize that security layers are not interchangeable, as each serves a distinct defensive purpose. It also allows students to identify potential imbalances in their existing security practices, particularly with an emphasis on preventive measures relative to detection and recovery mechanisms. However, this inventory is organized primarily from the defender's perspective by focusing on existing safeguards, whereas an attacker typically approaches a system by identifying exploitable assets, vulnerabilities, and weaknesses. This distinction provides the motivation for Stage III, which introduces threat-oriented analysis.

C. Stage III: Threat Modeling

Stage III shifts the perspective from the defender to the attacker by introducing a threat-oriented analysis. Students examine a selected system by addressing four fundamental principles of professional threat modeling [15]:

- What asset requires protection?
- What threats could compromise the asset?
- What countermeasures are currently in place to mitigate these threats?
- What security gaps or single points of failure remain?

1) Choice of method: Without structured guidance, novice learners may identify only a limited set of familiar attacks when addressing the second principle. An established threat-modeling method can provide a systematic framework for identifying a broader range of threats. Four approaches were considered. PASTA follows a seven-stage, risk-centric process that typically requires organizational and business context beyond the experience of novice learners [17]. DREAD provides quantitative risk scores, but the subjectivity associated with assigning these scores has been identified as a limitation [18]. STRIDE, in contrast, organizes threats into six mnemonic categories [15] that can be accessible concept for learners, while attack trees [16] provide a visual representation of alternative attack paths and potential points of failure. STRIDE is selected as the primary framework for systematically identifying relevant threats, while attack trees were incorporated to examine security gaps and potential single points of failure. To maintain accessibility for secondary-level learners, risk evaluation is simplified to qualitative high, medium, or low ratings based on the estimated likelihood and potential impact of each threat [38].

2) STRIDE prompts: Table III presents the student-facing prompt associated with each STRIDE category, an illustrative example drawn from an everyday context, and the corresponding threat identified in the hardware security case introduced in Stage IV. Rather than requiring students to memorize the STRIDE category names, the prompts serve as a structured checklist for systematically considering a broader range of potential threats beyond those that may initially come to mind. The hardware-security mappings presented in the final column were developed specifically for the proposed instructional framework and are revisited in Stage IV to demonstrate how threat-modeling concepts can be transferred from familiar cybersecurity scenarios to hardware security contexts.

3) Attack trees: To address the fourth principle, students construct a simplified attack tree in which the attacker's objective is represented as the root, alternative attack paths as branches, and existing security layers as controls associated with the branches they are intended to mitigate (Fig. 4). The attack tree is then used to evaluate the DiD design according to two criteria. First, any branch lacking a protective layer indicates a potential single point of failure. Second, multiple protections concentrated on the same branch reflect redundancy rather than defensive diversity. A more resilient design distributes preventive, detective, containment, and recovery mechanisms across distinct attack paths [31], [38].

Following the cognitive apprenticeship approach, the teacher first models a complete analysis, then guides groups through the analysis of a second system using the worksheet,

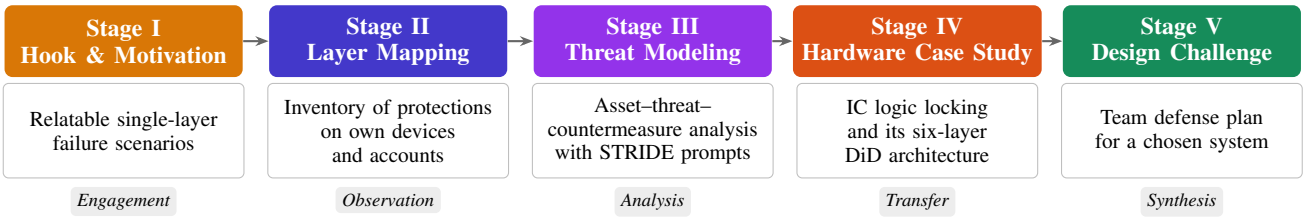


Fig. 3. Five-stage instructional framework for teaching Defense-in-Depth.

TABLE II. FIVE-STAGE FRAMEWORK: GUIDING QUESTION, PRINCIPLE, OUTPUT, AND THEORETICAL BASIS

Stage	Focus	Guiding question	DiD principle	Student output	Theoretical basis and how it is applied
I	Hook and Motivation	What happens if the only layer fails?	Single point of failure	Short reflection	Inquiry-based learning, orientation phase [21]: a puzzling failure creates the question that the module answers, and activates prior knowledge [20].
II	Layer Mapping	Which protections already exist, and what does each one do?	Independent, diverse layers	Layer inventory	Constructivism and the first half of Kolb's cycle [24]: students start from concrete experience with their own devices and reflect on it.
III	Threat Modeling	What are the asset, the threats, the countermeasures, and the gaps?	Asset–threat alignment	Threat worksheet and attack tree	Cognitive apprenticeship [22], [23]: the teacher models expert reasoning aloud, the worksheet and prompt card scaffold it, and support fades across examples.
IV	Hardware Case Study	Does the same reasoning protect an IC?	Concept transfer	Case response	Analogical transfer [25], [26]: comparing two structurally similar cases (account, ICs) leads students to abstract the DiD schema.
V	Design Challenge	How can we build a layered defense ourselves?	Applied systems thinking	Team defense plan	Experiential learning, active experimentation [24], carried out in teams so that peers provide support within the zone of proximal development [20].

TABLE III. STUDENT-FACING ADAPTATION OF STRIDE USED IN STAGE III AND REVISITED IN STAGE IV

STRIDE category	Prompt given to students	Everyday example	Typical defensive layer	Counterpart in the IC case (Stage IV)
Spoofing	Could someone pretend to be me or a service I trust?	Fake school login page	MFA; phishing awareness	Cloned or overproduced ICs sold as genuine
Tampering	Could someone change the data or the device?	Malware altering files	Patching; antivirus; backups	Hardware Trojan inserted during fabrication
Repudiation	Could someone act without leaving a trace?	Unnoticed login from a new device	Login alerts; activity logs	Probing attempt that goes undetected
Information disclosure	Could someone see what they should not?	Reused password leaked in a breach	Unique passwords; encryption	Key recovered by reverse engineering, probing, or SAT attack
Denial of service	Could someone stop me from using it?	Ransomware; lost phone	Backups; recovery codes	Trojan that disables the ICs
Elevation of privilege	Could someone obtain more access than they are allowed?	Student account gaining administrator rights	Least privilege; role-based permissions	Test (scan) access misused to read internal state

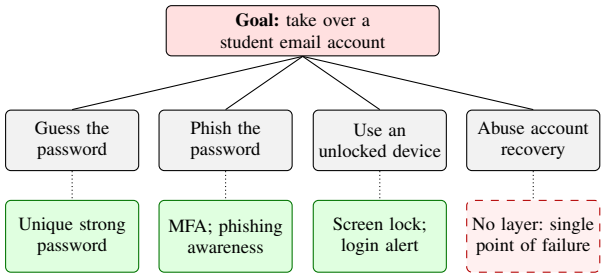


Fig. 4. Example of the simplified attack tree drawn in Stage III. Shaded boxes on the lower row are existing layers; the dashed box marks an unprotected branch.

and finally allows groups to analyze a third system using only the prompt card. At this stage, students have developed a structured method for analysis. Stage IV examines whether this method can be transferred to a different domain.

D. Stage IV: Hardware Security Case Study

The case study is intentionally concise, as its primary purpose is to facilitate knowledge transfer. Students are introduced only to the hardware concepts necessary to understand and apply the comparison.

1) The problem: Integrated circuits (ICs) are often designed by one organization and fabricated, packaged, and tested by external foundries and service providers, which may expose the design to potentially untrusted entities within the semiconductor supply chain. Such exposure can create risks including intellectual-property theft, unauthorized overproduction, reverse engineering, and the insertion of malicious circuitry known as hardware Trojans [5], [39]. *Logic locking* is one hardware-security technique used to protect circuit functionality and intellectual property by introducing key-controlled logic that enables the IC to operate correctly only when the appropriate secret key is applied [40], [41]. For instructional purposes, this mechanism can be compared to a machine that operates only when the correct combination is

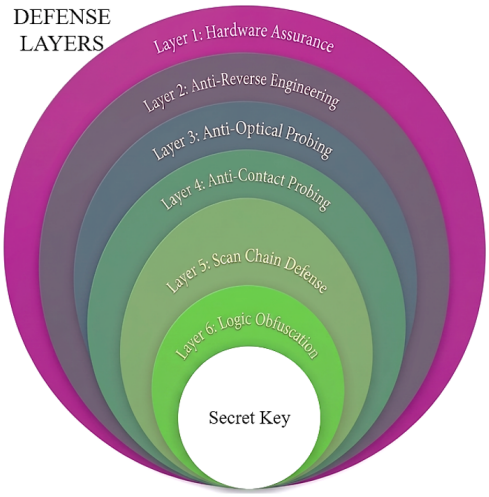


Fig. 5. Defense-in-Depth layers surrounding the secret key of a logic-locked integrated circuit (adapted from [5]).

entered. In this analogy, the secret key represents the asset that requires protection.

2) Why one lock is insufficient: An adversary may attempt to recover the key through multiple attack paths. These include supplying inputs to a functioning IC and inferring the key from its outputs (oracle or SAT attacks), reconstructing the circuit through physical analysis, observing internal signals optically without direct contact, probing internal wires, or exploiting scan chains intended for testing [5], [42], [43]. A defense that protects against one attack path may not prevent others. This reflects the same principle students observed with passwords in Stage I.

3) The six layers: Rahman *et al.* [5] arrange six defenses around the key, as in Fig. 5. Table IV pairs each layer with the attack it defeats and with an everyday analogue, and it is the only hardware material students must work with. Representative techniques are Trojan detection for hardware assurance [39], camouflaging for anti-reverse engineering [40], optical-probing detection [44], scan obfuscation [42], [43], and SAT-resistant locking such as SARLock, Anti-SAT, and SFLL [45]–[47]. Students often do not study how these techniques work.

4) Classroom activity: Groups receive the designated asset (the key), together with shuffled cards representing five attack paths and six defensive layers. They apply the four analytical prompts and STRIDE-based guidance introduced in Stage III, position each defensive layer on the attack path it is intended to mitigate, and compare the resulting analysis with Fig. 4 and Table III. The concluding discussion introduces risk-proportional reasoning: a personal account, a school network, and an integrated circuit may warrant different levels of security investment, although the underlying analytical process remains consistent. After applying the same method to two distinct systems, students are expected to use it independently.

E. Stage V: Design Challenge

Learners as a team apply DiD reasoning to a system of their choice, such as a student email account, a school network, a smart-home device, an online banking portal, or a hospital database. Each team must 1) identify the asset and explain its importance, 2) list at least four threats drawn from at least three STRIDE categories, 3) propose one or more defensive layers for each threat and classify the function of each layer, and 4) explain the consequences of the failure of any single layer. The emphasis is on the rationale behind the design rather than on the technically perfect mechanisms. The teams present their plans to the class and the plans are evaluated using the rubric described in Section VI.

VI. Assessment

Assessment measure students' ability to define DiD. It determines whether learners can identify assets and threats, select diverse defensive layers, and reason about potential failures. Table V presents the five assessment instruments, the objectives it measures, and the performance criterion for each. Formative assessments evaluate understanding throughout Stages I–IV, while summative assessments determine whether students can apply DiD reasoning to a new system.

1) Analytic rubric: Table VI provides the rubric used for the design task and for scenario analysis. It is analytic instead of holistic, since analytic rubrics with explicit level descriptors improve scoring consistency and give students usable feedback [48], [49]. The four content criteria correspond to the four threat-modeling questions, and a fifth addresses communication. Scores range from 5 to 20. A total of at least 15 with no criterion at level 1 is taken as meeting the module objectives. Teachers may adjust the assessment weights and performance thresholds to align with assessment policies.

2) Evaluation metrics: For teachers and for the planned pilot, the instruments yield five quantitative indicators: the quiz score, the number of distinct STRIDE categories in a threat list, the number of defensive functions covered by a plan, the count of unprotected branches remaining in an attack tree, and the rubric with its per-criterion profile.

VII. Classroom Implementation

A. Preparation and Resources

Table VII presents the recommended implementation plan. The complete intermediate course requires 15 modules with 120–180-minutes to complete each module. When instructional time is limited, Stages I and II can be combined into a single period, and Stage IV may be reduced to the card-based activity. All activities is designed to be conducted in learning management systems using shared digital document.

1) Prerequisites: Participating teachers and their students are expected to have basic digital literacy and experience. No prior background in programming, networking, or electronics is required. Teachers enrolled in intermediate module were expected to complete PERFECT Tier 1 or possess equivalent foundational knowledge. Stage IV requires only a brief introduction and does not assume any prior expertise in semiconductor technology.

2) Classroom management and workload: Group work in Stages III–V can be managed more effectively by assigning specific roles, such as asset owner, attacker, defender, and recorder, and rotating these roles between stages so that

TABLE IV. DEFENSE-IN-DEPTH LAYERS: HARDWARE SECURITY AND EVERYDAY ANALOGUES

Layer	Hardware defense (IC)	Attack defeated	Everyday analogue
1. Hardware assurance	Trojan detection through imaging and inspection	Trojan insertion	Verifying software authenticity; trusted sources
2. Anti-reverse engineering	Camouflaged cells; dummy vias; layout obfuscation	Reverse engineering	Code obfuscation; license enforcement
3. Anti-optical probing	Optical scattering; opaque layers; sensors	Optical (contactless) probing	Encryption of data in transit
4. Anti-contact probing	Active wire shield; probe-attempt detector	Electrical probing	Physical server security; tamper-evident seals
5. Scan-chain defense	Encrypted or dynamically obfuscated scan chains	Scan-chain attacks	MFA; role-based permissions
6. Logic obfuscation	SARLock, Anti-SAT, SFLL logic locking	SAT and oracle attacks	Strong passwords; password managers

TABLE V. ASSESSMENT INSTRUMENTS, OBJECTIVES, AND PERFORMANCE CRITERIA

Instrument	Type	LOs	Evidence and performance criterion	Weight
Concept quiz	Form.	1, 2	8–10 items on definitions, layer functions, and diversity versus redundancy; criterion: 70% correct	15%
Scenario analysis	Summ.	3, 6	Written breach narrative; student names the failed or missing layer and proposes a layer with a *different* function; rubric criteria C2–C4	20%
Hardware case response	Form.	4	Six layer–attack pairs matched and one analogy explained; criterion: five of six pairs correct	15%
Layered design task	Summ.	5, 6	Team defense plan from Stage V; full rubric (Table VI)	35%
Reflection writing	Form.	1	One personal system, one gap found, one action planned; scored for completion	15%

Note: LOs. = Learning Objectives; Form. = formative; Summ. = summative.

TABLE VI. ANALYTIC RUBRIC FOR THE DEFENSE-IN-DEPTH DESIGN TASK (5–20 POINTS)

Criterion	1 – Beginning	2 – Developing	3 – Proficient	4 – Advanced
C1. Asset identification	Asset missing, or confused with a device or tool	Asset named, but its value or owner is unclear	Asset named, with why it matters and to whom	Also separates the primary asset from supporting assets (e.g., the recovery email)
C2. Threat analysis	Fewer than two threats, or threats unrelated to the asset	Two or three threats, all from one STRIDE category	At least four specific threats from three or more STRIDE categories	Also includes human and physical threats and a realistic attacker capability
C3. Layer–threat alignment and diversity	Protections listed without links to threats	Every threat has a layer, but layers repeat one function	Every threat mapped to a layer; layers cover at least three of prevent, detect, contain, recover	Also justifies layers by asset value and cost, and removes redundant layers
C4. Failure-path reasoning	Not addressed	States that other layers "help" without tracing a path	For at least two layers, traces what the attacker gains on failure and which layer acts next	Also finds a remaining single point of failure and proposes a fix
C5. Communication of rationale	Plan cannot be followed	Understandable, but choices are not justified	Clear diagram or table with a rationale for each choice	Also uses correct vocabulary; a peer could implement the plan

TABLE VII. RECOMMENDED IMPLEMENTATION PLAN (45–50 MINUTE PERIODS)

Stage	Periods	Teacher preparation and materials	Grouping
I	1	Select two or three recent, age-appropriate incidents; prepare discussion prompts	Whole class; think–pair–share
II	1	Layer-inventory sheet; check school policy on personal devices	Pairs
III	2	Rehearse one worked example; threat worksheet, STRIDE prompt card, chart paper	Groups of 3–4
IV	1–2	Read the case brief (Fig. 5, Table IV); print attack and layer cards	Groups of 3–4
V	2–3	Plan template; rubric handed out in advance; presentation schedule	Teams of 3–4 with roles

each student has an opportunity to practice the attacker's perspective. Preparation effort is concentrated primarily in Stages III and IV, where the teacher should be familiar with at least one worked example. Grading effort is concentrated in Stage V, where the analytic rubric enables the instructor to complete a single scoring pass for each team rather than for each individual student.

B. Pedagogical Challenges and Mitigations

1) Abstraction: Concepts such as threats, attack surfaces, and defensive layers can be abstract and difficult for novice learners to visualize [3]. The module therefore begins with familiar physical analogies, such as a locked door or a vault, before progressing to cybersecurity systems, and uses visual representations to reinforce these concepts (Fig. 1, Fig. 4).

2) Technical overload: Logic locking, optical probing, and SAT attacks extend beyond the scope of the secondary-level curriculum. The case study is therefore limited to one figure, one table, and a card-based activity, and students are assessed on their ability to match defensive layers to corresponding attacks rather than on the underlying technical mechanisms.

3) Differentiated skill level: Students may enter the module with varying levels of computing experience. Teachers can therefore adjust the complexity of the selected system: beginners may analyze a personal account, while more advanced students may examine an IC or a hospital database in Stage V [7].

4) Perceived irrelevance: Some students may not initially perceive cybersecurity as personally relevant. Stage I therefore introduces familiar threats, such as password reuse, phishing, lost devices, and failed account recovery, to connect cybersecurity concepts with students' everyday experiences.

VIII. Limitations and Evaluation Plan

The primary limitation of this work is that it presents a design framework; an empirical evaluation has yet to be conducted. Therefore, its effectiveness remains to be empirically established in implementation. The stage durations presented in Table VII and the rubric thresholds are design recommendations that may be refined based on findings from future pilot implementations. The hardware mapping in Table III is intended as an instructional aid rather than a formal threat model of logic locking. Similarly, everyday analogies in Table IV are approximated.

The next phase of this research [6] will involve a classroom pilot to examine three hypotheses:

- H1. After completing the module, learners will identify more distinct threat categories and defensive functions for a given system than before completing the module.
- H2. After completing the module, learners will analyze an unfamiliar system not used during instruction at rubric level 3 or above on criteria C2-C4.
- H3. Two teachers independently scoring the same student plans using Table VI will achieve at least moderate inter-rater agreement (Cohen's $\kappa \geq 0.6$).

The pilot study will employ a single-group pretest-posttest design using the concept quiz and a scenario-analysis task as pre- and post-measures, together with the indicators described in Section VI, student feedback, and teacher observation logs documenting the actual time required for each stage. Future work will adapt the module for introductory and advanced courses and explore interactive and gamified implementations of the five-stage framework [50].

IX. Conclusion

Defense-in-Depth helps learners understand security as a layered and resilient system rather than as a single protective mechanism. This study presents a five-stage framework for teaching DiD in high school. In response to DQ1, the sequence progresses from a motivating failure scenario through layer mapping and STRIDE-supported threat modeling to an independent design task, with each stage grounded in an established learning theory and aligned with measurable objectives. In response to DQ2, the logic-locking case is reduced to one concept, one table, and one card-based activity so that it functions primarily as a transfer learning task. In response to DQ3, five assessment instruments and a five-criterion analytic rubric make asset identification, threat analysis, layer diversity, and failure-path reasoning observable and assessable. The framework has not yet been evaluated with pilot study. The intended outcome is that students learn to ask not only whether a system is protected, but also how it remains secure when an individual protective measure fails.

Acknowledgment

This work was supported by the United States Office of Naval Research (ONR) under Grant No. N00014-23-1-2454 through the PERFECT (Providing Environment and Resources for STEM 9th–12th Graders in Effective Cybersecurity Training) project at the University of North Texas. The authors thank the participating STEM educators from Amarillo ISD, Boyd ISD, Bushland ISD, Canyon ISD, and Denton ISD for their engagement and feedback on the course modules, and the faculty members, graduate researchers, and student assistants at the University of North Texas and West Texas A&M University who contributed to curriculum development, instructional delivery, and project coordination.

References

[1] Z. Whittaker, "Instructure strikes deal with hackers who breached it twice," TechCrunch, May 2026, [Online]. Available: https://techcrunch.com/2026/05/12/instructure-strikes-deal-with-hackers-who-breached-it-twice/.

[2] S. P. Mohammed, G. Hossain, and S. Y. Quadri Ameen, "Cybersecurity data visualization: Designing a course for future high school students," in *Proc. 12th Int. Symp. Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, Apr. 2024, pp. 1–7.

[3] D. Pencheva, J. Hallett, and A. Rashid, "Bringing cyber to school: Integrating cybersecurity into secondary school education," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 68–74, Mar. 2020.

[4] H. Son and S. Kim, "Defense-in-depth strategy for smart service server cyber security," in *Computer Applications for Communication, Networking, and Digital Contents*. Berlin, Germany: Springer, 2012, vol. 350, pp. 181–188.

[5] M. T. Rahman *et al.*, "Defense-in-depth: A recipe for logic locking to prevail," *Integration*, vol. 72, pp. 39–57, May 2020.

[6] Design-Based Research Collective, "Design-based research: An emerging paradigm for educational inquiry," *Educ. Researcher*, vol. 32, no. 1, pp. 5–8, 2003.

[7] M. Kamaludeen, S. Ismaeel, S. Asiri, T. Allen, and C. Scarfo, "A framework for cyber protection (FCP) in K–12 education sector," *IET Conf. Proc.*, vol. 2020, no. 6, pp. 239–244, Jun. 2021.

[8] P. Vajpayee and G. Hossain, "Cybersecurity education in high school: Exploring cyber assets, cyber value at risk, and authentic assessment," in *Proc. IEEE Frontiers in Education Conf. (FIE)*, Washington, DC, USA, Oct. 2024, pp. 1–9.

[9] A. Ibrahim, M. McKee, L. F. Sikos, and N. F. Johnson, "A systematic review of K–12 cybersecurity education around the world," *IEEE Access*, vol. 12, pp. 59 726–59 738, 2024.

[10] M. Nusrat, S. Bhuiyan, and G. Hossain, "Don't Just Teach, Explain! A Gamified 20Q Recommender for Cybersecurity Education", *arXiv preprint arXiv:2604.26964*, 2026.

[11] R. B. Sağlam, V. Miller, and V. N. L. Franqueira, "A systematic literature review on cyber security education for children," *IEEE Trans. Educ.*, vol. 66, no. 3, pp. 274–286, 2023.

[12] A. A. Arishi, N. H. Kamarudin, K. A. A. Bakar, Z. B. Shukur, and M. K. Hasan, "Cybersecurity awareness in schools: A systematic review of practices, challenges, and target audiences," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 12, pp. 467–473, 2024.

[13] Y. Xu and H. Li, "Cybersecurity matters for primary school students: A scoping review of the trends, challenges, and opportunities," *IEEE Trans. Learn. Technol.*, vol. 18, pp. 513–529, 2025.

[14] M. Nusrat, S. Bhuiyan, and G. Hossain, "Securing the Cognitive Enterprise: Zero-Trust Agentic Knowledge Retention in Cyber-Physical Systems", in *Proc. 7th Int. Conf. Artificial Intelligence, Robotics, and Control (AIRC)*, Savannah, GA, USA, 2026, pp. 421–427, doi: 10.1109/AIRC69745.2026.11631472.

[15] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.

[16] B. Schneier, "Attack trees," *Dr. Dobb's J.*, vol. 24, no. 12, pp. 21–29, Dec. 1999.

[17] T. UcedaVélez and M. M. Morana, *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hoboken, NJ, USA: Wiley, 2015.

[18] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody, "Threat modeling: A summary of available methods," Software Engineering Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA, White Paper, Jul. 2018.

[19] S. S. Joshi, P. Mukherjee, K. A. Davis, and J. C. Davis, "Introducing systems thinking as a framework for teaching and assessing threat modeling competency," in *Proc. ASEE Annu. Conf. & Expo.*, 2024, arXiv:2404.16632.

[20] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA, USA: Harvard Univ. Press, 1978.

[21] M. Pedaste *et al.*, "Phases of inquiry-based learning: Definitions and the inquiry cycle," *Educ. Res. Rev.*, vol. 14, pp. 47–61, 2015.

[22] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics," in *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*, L. B. Resnick, Ed. Hillsdale, NJ, USA: Lawrence Erlbaum, 1989, pp. 453–494.

[23] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *J. Child Psychol. Psychiatry*, vol. 17, no. 2, pp. 89–100, 1976.

[24] D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1984.

[25] M. L. Gick and K. J. Holyoak, "Schema induction and analogical transfer," *Cogn. Psychol.*, vol. 15, no. 1, pp. 1–38, 1983.

[26] D. N. Perkins and G. Salomon, "Transfer of learning," in *International Encyclopedia of Education*, 2nd ed. Oxford, U.K.: Pergamon, 1992.

[27] B. S. Bloom *et al.*, *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York, NY, USA: David McKay, 1956.

[28] L. W. Anderson and D. R. Krathwohl, Eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY, USA: Longman, 2001.

[29] M. Nusrat, H. R. Mahi, S. Bhuiyan, and G. Hossain, "Mitigating the Information Cocoon Effect in Cognitively Aligned Recommendations: A Human-Centered Approach", in *Proc. IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2026, pp. 324–330, doi: 10.1109/CCWC67433.2026.11393796.

[30] T. Abdelghani, "Implementation of defense in depth strategy to secure industrial control system in critical infrastructures," *Amer. J. Artif. Intell.*, vol. 3, no. 2, pp. 17–24, 2019.

[31] R. Ross, M. McEvilley, and J. Carrier Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," NIST, Special Publication 800-160, Nov. 2016.

[32] PERFECT Project, "Providing environment and resources for STEM 9th–12th graders in effective cybersecurity training," University of North Texas, 2024, [Online]. Available: https://ci.unt.edu/perfect-project/index.html.

[33] M. Nusrat, T. Raheem, M. Shin, and G. Hossain, "Bridging Industry Standards with the PERFECT Project: A Foundational Cybersecurity Education in High Schools", *Preprint*, University of Bahrain, Jul. 16, 2026. [Online]. Available: https://journal.uob.edu.bh/items/3db81670-3650-4cd9-b03d-2618ec328db4

[34] N. Mary and G. Hossain, "Towards Personalized Recommender System: A Gray-Box Modeling Approach," in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2025, pp. 1–6.

[35] Computer Science Teachers Association, "CSTA K–12 computer science standards, revised 2017," 2017, [Online]. Available: https://www.csteachers.org/standards.

[36] R. Petersen, D. Santos, M. C. Smith, K. A. Wetzel, and G. Witte, "Workforce framework for cybersecurity (NICE framework)," NIST, Special Publication 800-181 Rev. 1, Nov. 2020.

[37] T. Raheem, M. Nusrat, and G. Hossain, "A SWOT analysis of mobile privacy and security education for K–12 students," in *Proc. 14th Int. Symp. Digital Forensics and Security (ISDFS)*, Boston, MA, USA, 2026, pp. 1–6.

[38] Joint Task Force Transformation Initiative, "Guide for conducting risk assessments," NIST, Gaithersburg, MD, USA, Special Publication 800-30 Rev. 1, 2012.

[39] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan. 2010.

[40] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proc. 49th Annu. Design Automation Conf.*, San Francisco, CA, USA, Jun. 2012, pp. 83–89.

[41] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)*, Washington, DC, USA, May 2015, pp. 137–143.

[42] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure scan and test using obfuscation throughout supply chain," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 9, pp. 1867–1880, Sep. 2018.

[43] A. Cui, Y. Luo, and C.-H. Chang, "Static and dynamic obfuscations of scan data against scan-based side-channel attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 363–376, Feb. 2017.

[44] S. Khalaj Monfared, K. Mitard, A. Cannon, D. Forte, and S. Tajik, "LaserEscape: Detecting and mitigating optical probing attacks," in *Proc. 43rd IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, New York, NY, USA, Oct. 2024, pp. 1–10.

[45] M. Yasin, B. Mazumdar, J. J. V. Rajendran, and O. Sinanoglu, "SARLock: SAT attack resistant logic locking," in *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)*, McLean, VA, USA, May 2016, pp. 236–241.

[46] Y. Xie and A. Srivastava, "Anti-SAT: Mitigating SAT attack on logic locking," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 38, no. 2, pp. 199–207, Feb. 2019.

[47] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, Dallas, TX, USA, Oct. 2017, pp. 1601–1618.

[48] A. Jonsson and G. Svingby, "The use of scoring rubrics: Reliability, validity and educational consequences," *Educ. Res. Rev.*, vol. 2, no. 2, pp. 130–144, 2007.

[49] S. M. Brookhart, *How to Create and Use Rubrics for Formative Assessment and Grading*. Alexandria, VA, USA: ASCD, 2013.

[50] M. Nusrat, G. Hossain, Kinshuk, and S. Bhuiyan, "Learning-to-explain through 20Q gaming: An explainable recommender for cybersecurity education," in *Proc. IEEE 16th Annu. Information Technology, Electronics and Mobile Communication Conf. (IEMCON)*, Berkeley, CA, USA, 2025, pp. 381–388.